Dolog AKF → A350/A500
Type: AKF35EN
Version 6.4

Programming Instructions
DOK-276529.22-0793

Translation of the German Description DOK-275576

Belongs to software kit E-No. 424-271510

Notes to AKF35EN, Version 6.42
Programming Instructions No. DOK-276529.22

Backup with DOS 6.x

From DOS version 6.0 upwards the file ”backup.exe” will no longer be installed.

☞ **Note** Together with DOS 6.2 diskettes an additional diskette will be provided with the file ”backup.exe”.

DOS 6.0 / 6.2 allows the use of the backup file from former version (e.g. DOS 5.0). This must be handled in the following 3 steps:

- Enter ”devicehigh = C:\dos\dosver.exe into config.sys file.
- Input of the requested DOS version by the command ”C:\dosver\ ......”
- Reboot DOS

☞ **Note** It is to prefer to execute the backup by special software. This way an independancy from the DOS version will be achieved.

P/N: 33001452.01

AKF35(V641) with Windows95/98, Windows NT, OS/2

⚠ **Caution AKF35 is only guaranted for DOS operation systems.**

Any how we have no information that AKF35 will not run with other operation systems.

☞ **Note** With these operation systems the backup function will not work.

12 / 97:
I Notes to ”PADT on Modnet 2/NP” together with AKF35

Warning If you drive your PADT on Modnet 2/NP, the PADT can crash in the following cases:

- **when attempting to include a Modent 2/NP networking if the RS 232 connection is still active at the same time.**
  Remedy:
  Set the type of networking to Networking = None in the menu ”Setup → Networking” and remove the connections PADT ↔ PLC before starting the Modnet 2/NP networking**.**

- **when releasing an active Modnet 2/NP networking (i.e. Networking = None)**
  Remedy:
  No release of the networking or release of networking after terminating work on your PADT.
  In this case you must reboot the PADT if you want to continue work.

Note You will find some information about working with the software AKF35EN, Version 6.4 in files with the extension .DOC or .TXT on the second AKF35EN system disk.

II Modicon A350/A500 with BSW <V5.06 and <V6.01 together with AKF35 V6.x: Bsdol function RAMZU-SEAB

When programming or operating your PLC with AKF35 on Modnet 2/NP please note the following:

Caution If you have carried out the first use of PLC with AKF35 V6.x you may not use for other purposes the free segment area which is located in front of the area used from RAMZU-SEAB. Otherwise you risk a failure of the perfomance ”AKF on bus”. (With AKF35 V5.x generally a complete segment is used for RAMZU-SEAB, that means no change is necessary.)

Note This failure does not occur any more with basic software version 5.06 or 6.01.

Remedy:
If you need this free area urgently, you have to redefine the area for RAMZU-SEAB online on the first address of this segment. The now following free area can be used for other purposes.

For this proceed as the following (entries are underlined):

Step 1 ”Online” → ”Terminal Mode”

Reaction On the screen appears:
`Dolog B:`

Step 2 Enter the Bsdol function
`RAMZU-SEAB`.

Step 3 Following the requests on the screen you define a new segment area beginning at address 0 of one segment:

Example:
After first use of PLC with AKF35 V6.x the allocation for RAMZU-SEAB is as following:
`15:18432, 4095.`
With following entry you redefine this area at the begin of segment 15:
`15:0, 4095`.

☞ **Note** This change is **not possible during active** communication.

Step 4 Change the memory allocation defined during first use of PLC with the Bsdol function ASB (Change and Diplay Memory Area)

Example: (Entries are underlined)

```
DOLOG B:   ASB <RETURN>
NUMBER OF STORAGE AREA    :30 <RETURN>
STORAGE AREA 30(RAM) 15: 18433 TO 22528
SEGMENT: 15 <RETURN>
FRM: 1 <RETURN>
TO: 4096 <RETURN>
TYPE=RAM
STORAGE AREA 31      : E <RETURN>
```

React. Now you can use the free area beyond the end of the RAMZU-SEAB area for any other purpose.

III Modicon A500 with BSW V6.x and AKF35 V6.x Modification of Equipment List when using Intelligent Function Modules

Warning **You only may modify the equipment list offline with AKF35.**
An online modification with Bsdol function BES ”Enter and Change Equipment List” is not allowed, the PLC can crash in this case.

IV Reservation of Areas for Parameter Fields in the Cross-Reference List (AKF35)

Note If the length of a parameter field is not known exactly during configuration, the system reserves an area of max. length and will display this in the cross-reference list.

For fields whose length cannot be defined exactly or for which the field length is >255, only the start and possibly the end of the field are entered in the cross-reference list as single signal(s). Since the cross-reference list is incomplete in this case, a respective message is given.

Example: SFB No. 259 (AWE13)

The parameter WA is the first word of a field, in which the input of the measured values is stored. The number of channels is only computed at PLC runtime out of the parameter KA. The maximum value of KA is 255, that means a reservation of 255 maker words starting with address WA is done by the system.

V Counter and Timer in the same network: Malfunction in AKF35 Version 6.41

(12/97)

Problem
If you place the output of a timer onto the reset input of a counter in the same network, the counter will take the setpoint value of the timer as setpoint value rather than its own programmed setpoint value. For instance, if you entered a setpoint value of 10 for the timer, the counter will set its output after 10 pulses, regardless of the setpoint value indicated in its own makrer word.

Remedy
This malfunction will be prevented if you program timer and counter in different networks.

Notes to AKF35EN, Version 6.4
Programming Instructions No. DOK-276529.22

6 / 99:
Year 2000 Compatibility (Y2k):

The Y2k correction are realized from now software version 6.4 but restricted by two minor documentation issues. These functions are:
– LOAD / COMPARE / BLOCKS WITH PLC und
– SPECIAL / DIRECTORY / AKF–BLOCKS.
Year 2000 will be printed ore displayed as 100. All further years will be shown accordingly.

Backup with DOS 6.x

From DOS version 6.0 upwards the file ”backup.exe” will no longer be installed.

☞ **Note** Together with DOS 6.2 diskettes an additional diskette will be provided with the file ”backup.exe”.

DOS 6.0 / 6.2 allows the use of the backup file from former version (e.g. DOS 5.0). This must be handled in the following 3 steps:

❒ Enter ”devicehigh = C:\dos\dosver.exe into config.sys file.

❒ Input of the requested DOS version by the command ”C:\dosver\ ......”

❒ Reboot DOS

☞ **Note** It is to prefer to execute the backup by special software. This way an independancy from the DOS version will be achieved.

P/N: 33001452.00

AKF35(V641) with Windows95/98, Windows NT, OS/2

Caution **AKF35 is only guaranted for DOS operation systems.**

Any how we have no information that AKF35 will not run with other operation systems.

Note With these operation systems the backup function will not work.

12 / 97:
I Notes to ”PADT on Modnet 2/NP” together with AKF35

Warning **If you drive your PADT on Modnet 2/NP, the PADT can crash in the following cases:**

– **when attempting to include a Modent 2/NP networking if the RS 232 connection is still active at the same time.**
  Remedy:
  Set the type of networking to Networking = None in the menu ”Setup → Networking” and remove the connections PADT ↔ PLC before starting the Modnet 2/NP networking**.**

– **when releasing an active Modnet 2/NP networking (i.e. Networking = None)**
  Remedy:
  No release of the networking or release of networking after terminating work on your PADT.
  In this case you must reboot the PADT if you want to continue work.

Note You will find some information about working with the software AKF35EN, Version 6.4 in files with the extension .DOC or .TXT on the second AKF35EN system disk.

II Modicon A350/A500 with BSW <V5.06 and <V6.01 together with AKF35 V6.x: Bsdol function RAMZU-SEAB

When programming or operating your PLC with AKF35 on Modnet 2/NP please note the following:

Caution If you have carried out the first use of PLC with AKF35 V6.x you may not use for other purposes the free segment area which is located in front of the area used from RAMZU-SEAB. Otherwise you risk a failure of the perfomance ”AKF on bus”. (With AKF35 V5.x generally a complete segment is used for RAMZU-SEAB, that means no change is necessary.)

Note This failure does not occur any more with basic software version 5.06 or 6.01.

Remedy:
If you need this free area urgently, you have to redefine the area for RAMZU-SEAB online on the first address of this segment. The now following free area can be used for other purposes.

For this proceed as the following (entries are underlined):

Step 1 ”Online” → ”Terminal Mode”

Reaction On the screen appears:
`Dolog B:`

Step 2 Enter the Bsdol function
`RAMZU-SEAB`.

Step 3 Following the requests on the screen you define a new segment area beginning at address 0 of one segment:

Example:
After first use of PLC with AKF35 V6.x the allocation for RAMZU-SEAB is as following:
`15:18432, 4095`.
With following entry you redefine this area at the begin of segment 15:
`15:0, 4095`.

Note This change is **not possible during active** communication.

Step 4 Change the memory allocation defined during first use of PLC with the Bsdol function ASB (Change and Diplay Memory Area)

Example: (Entries are underlined)

```
DOLOG B:  ASB <RETURN>
NUMBER OF STORAGE AREA    :30 <RETURN>
STORAGE AREA 30(RAM) 15: 18433 TO 22528
SEGMENT: 15 <RETURN>
FRM: 1 <RETURN>
TO: 4096 <RETURN>
TYPE=RAM
STORAGE AREA 31        : E <RETURN>
```

React. Now you can use the free area beyond the end of the RAMZU-SEAB area for any other purpose.

III Modicon A500 with BSW V6.x and AKF35 V6.x Modification of Equipment List when using Intelligent Function Modules

Warning **You only may modify the equipment list offline with AKF35.**
An online modification with Bsdol function BES ”Enter and Change Equipment List” is not allowed, the PLC can crash in this case.

IV Reservation of Areas for Parameter Fields in the Cross-Reference List (AKF35)

Note If the length of a parameter field is not known exactly during configuration, the system reserves an area of max. length and will display this in the cross-reference list.

For fields whose length cannot be defined exactly or for which the field length is >255, only the start and possibly the end of the field are entered in the cross-reference list as single signal(s). Since the cross-reference list is incomplete in this case, a respective message is given.

Example: SFB No. 259 (AWE13)

The parameter WA is the first word of a field, in which the input of the measured values is stored. The number of channels is only computed at PLC runtime out of the parameter KA. The maximum value of KA is 255, that means a reservation of 255 maker words starting with address WA is done by the system.

V Modicon A350/A500 with ALU 150 and AKF35, V6.x Program downloads into the PLC via quick PADT (P820C, P840C, IBM compatible PC with processor ≥ 486)

(05/94)

Caution When using a quick PADT and the menu item ”Load → Program to PC*” to download your AKF35 program into a PLC that is equipped with an ALU 150 central unit, the program will abort the transfer. The download operation remains locked in the function ”Initialization Equipment list front connection”, and the initial values are also not transferred.

Remedy
There are three alternatives to choose from:

- **Terminating the download operation after cancel**
  Proceed as follows to do this:

Step 1	Select the menu item ”Online → Terminal Mode”.
Reaction	The ALU 150 will be in the online function ”BES” (enter and modify equipment list).
Step 2	End the ”BES” function with the entry `E <Return>`.
Reaction	The screen will be show the prompt `Dolog B:`
Step 3	Exit the Terminal Mode.
Step 4	Transfer the initial values to the PLC through the menu item ”Load → Initial Value to PC*”.

- **Reducing the processor speed of PADT to perform the download operation as usual**
  Proceed as follows to do this:

Step 1 Invoke the BIOS setup program of your PADT (refer to the PADT Operating Instructions / System Manual).

Step 2 Set processor speed to ”Low”.

Step 3 Set the systems cache to ”OFF” which will disable it.

Step 4 End the BIOS setup program by saving the modified settings (refer to the PADT User’s Guide/Manual).

Reaction The PADT is now low enough to interact with the ALU 150 as usual.

☞ **Note** Return your PADT to its initial state when you have finished working with the AKF35/ALU150.

- **Utilizing a PADT with a slower processor (≤ 386)** for service with AKF35, Version 6.x, and Modicon A350/A500 with ALU150.

VI Counter and Timer in the same network: Malfunction in AKF35 Version 6.41

(12/97)

Problem

If you place the output of a timer onto the reset input of a counter in the same network, the counter will take the setpoint value of the timer as setpoint value rather than its own programmed setpoint value. For instance, if you entered a setpoint value of 10 for the timer, the counter will set its output after 10 pulses, regardless of the setpoint value indicated in its own makrer word.

Remedy

This malfunction will be prevented if you program timer and counter in different networks.

Overview

Notes
Table of Contents

Part I Getting Started

Part II Installation Instructions

Part III Programming Instructions

Part IV Formal Operands of SFBs

Part V Dolog AKF for Beginners

Part VI Index

Part VII Appendix

Part VIII

Part IX

Notes
Table of Contents

Notes

Application Note

Caution The relevant regulations must be observed for control applications involving safety requirements.
For reasons of safety and to ensure compliance with documented system data, repairs to components should be performed only by the manufacturer.

Training

AEG offers suitable training that provides further information concerning the system (see addresses).

Data, Illustrations, Alterations

Data and illustration are not binding. We reserve the right to alter our products in line with our policy of continuous product development. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us by using the form on the last page of this publication.

Addresses

The addresses of our Regional Sales Offices, Training Centers, Service and Engineering Sales Offices in Europe are given at the end of this publication.

Copyright
All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including copying, processing or any information storage, without permission in writing by the AEG Aktiengesellschaft. You are not authorized to translate this document into any other language.

Trademarks
All terms used in this user manual to denote AEG products are trademarks of the AEG Aktiengesellschaft.

IBM, IBM-PC, IBM-XT and IBM-AT are registered trademarks of International Business Machines Corporation.
Microsoft and MS-DOS are registered trademarks of Microsoft Corporation.

© 1993 AEG Aktiengesellschaft.

Terminology

Note This symbol emphasizes very important facts.

Caution This symbol refers to frequently appearing error sources.

Warning This symbol points to sources of danger that may cause financial and health damages or may have other aggravating consequences.

Expert This symbol is used when a more detailed information is given, which is intended exclusively for experts (special training required). Skipping this information does not interfere with understanding the publication and does not restrict standard application of the product.

Path This symbol identifies the use of paths in software menus.

Figures are given in the spelling corresponding to international practice and approved by SI (Système International d‘ Unités).
I.e. a space between the thousands and the usage of a decimal point (e.g.: 12 345.67).

Abkürzungen

ABS	Absolute addressing
Addr.	Address (signal address)
AKF	Instruction List, Ladder Diagram, Function Block Diagram
AWP	User program
CLC	Closed-loop Control
D-Word	Double word
FB	Function block
FBD	Function block diagram
F-Word	Floating point word
HW	Hardware (e.g. PLC)
IL	Instruction List
I/O	Input- / Outputsignals (e.g. of a module)
LD	Ladder diagram
OB	Organisation block
PADT	Programming and debugging tool (= programming panel)
PB	Program block
PLC / PC* (in SW)	Programmable Controller
React.	Reaction during description of steps (on screen)
SFB	Standard function block
SSP	Signal memory
SW	Software
SYM	Symbolic addressing
SYM/COM	Symbols and Comments
ZVT	Time management table
<Return>	Press the Return key
<Esc>	Press the Esc key
<Ctrl>+<Alt>+<Del>	Press the keys Ctrl, Alt and Del simultaneously (begin with Ctrl and end with Del)

Objectives

This description is intended for the person configuring user programs for the A350 and A500. Configuration is in Dolog AKF.
The person configuring is then able to

- set up the programming panel,
- install the software,
- configure with the software,
- document the written program,
- transfer the written program to the controller and to start it.

Furthermore, a system diagnosis can be made (dyn. status display).
The experienced AKF user will generally only use parts III, IV and VI as a reference manual.

Arrangement of This Guide

Notes This part contains the preface, the handling of diskettes and the complete table of contents of the manual.

Part I Getting Started
Here you learn what to do in what order in order to solve your automatic control problem for the A350 or A500.

Part II Installation Instructions
This part informs you about the requirements for the initial start-up of the programming panel and installation of the software.

Part III Programming Instructions

This part describes the use of the software and is designed for reference. Frequent use of the index in Part VI is therefore recommended.

Chapter 1	is a general introduction with new features of the software version
Chapter 2	describes the validity of Dolog AKF Software and the system markers of the A350/A500.
Chapter 3	explains the functions of the keyboard and the mouse. A detailed ”key table” can be found in the index.
Chapter 4	is the actual ”programming”. The menues are described in the windows according to their structure. The following order is valid: Edit, Load, Online, Print, Special, Setup. Some functions can be selected from different menues. Please use the index in part VI for orientation.

Part IV Formal Operands of SFBs

This part contains an alphabetical list of the standard function blocks. The formal operands are listed in a table and briefly explained for each SFB.

Part V AKF35 for Beginners

Part VI Index

Part VII Appendix

Addresses	contains the addresses of the sales offices domestically and abroad.
User comments	Please make frequent use of this form if you have suggestions or corrections to the documentation and software.

Related Documents

Modicon A350
Modular Automation Device
User’s Manual
A91V.12-234678

A500
Chassis Mount Controller
Standard Equipment
User Manual
A91M.12-279330

Dolog AKF Standard Function Blocks
(AKF35EN Version 5.x)
Block Library
A91M.12-271891

[1]A350/A500
Dolog AKF Standard-Funktionsbausteine
(AKF35 Version 6.x)
Bausteinbibliothek
A91M.12-279346

[1]A500
Grundsoftware Version 6.0
Benutzerhandbuch

A91M.12-279344

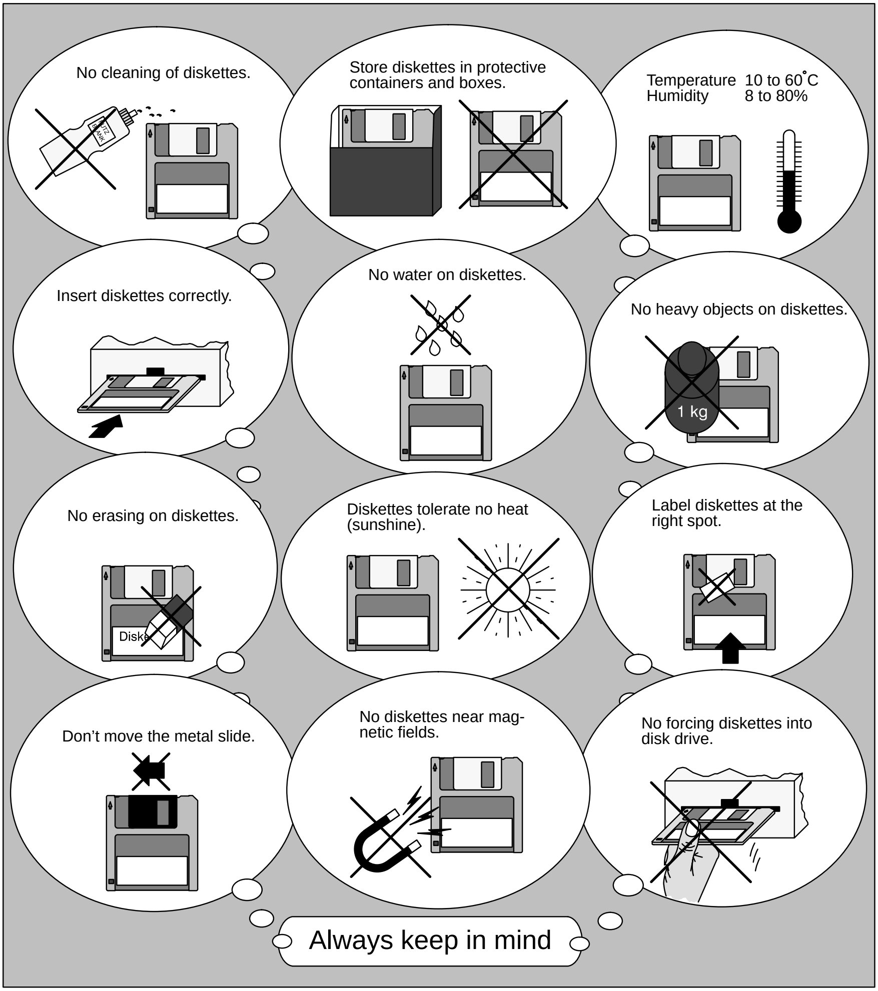
[1]A350/A500
Regeln mit Dolog AKF
Benutzerhandbuch
A91V.12-271963

Validity Note

This documentation is valid for Dolog AKF → A350/A500 as of version 6.0 in connection with basic software version 5.05 or as of version 6.0

1) in german language

Handling 3 $^1/_2$" Diskettes



Handling 5 $^{1}/_{4}$" Diskettes

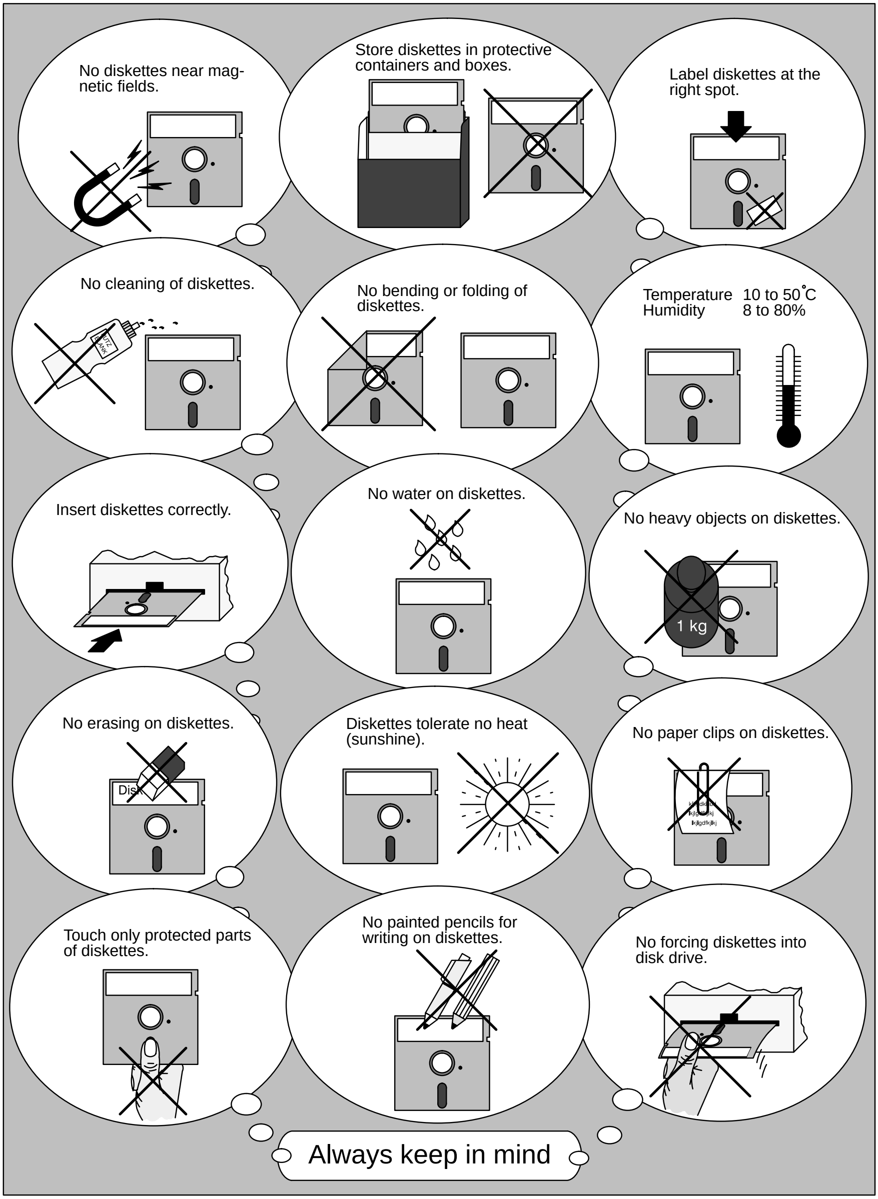


Table of Contents

Part I Getting Started 1

Chapter 1 Getting Started 3
1.1 General Information 4

Chapter 2 Checklists / Programming Sequence 5
2.1 Checklists 6
2.1.1 Installation 6
2.1.2 Configuration 6
2.1.3 Program Transmission/Start-up/Test 7
2.1.4 EPROM Operation 7
2.2 Programming Sequence 8
2.3 COMAKF Initialization in AKF 17

Part II Installation Instructions 19

Chapter 1 General 21
1.1 Summary 22

Chapter 2 Installation 23
2.1 Device selection/Requirements 24
2.1.1 Important Remarks 24
2.1.2 Requirements P510-40/P610 25
2.1.3 Requirements for IBM-compatible PLC 26
2.2 Creating a backup copy 27
2.3 Installation of the Software 29
2.3.1 Installation on the Hard Disk 29
2.4 Software Call Dolog AKF A350 / A500 30

Part III Programming Instructions 31

Chapter 1 Introduction 33
1.1 General 34
1.2 New Features of Versions 5.0 to 6.0 35
1.3 Basic Structure of the Software (V6.0) 38

Chapter 2 Validity Scopes and System Operands 41
2.1 Validity Scopes of the Software
Dolog AKF A350/A500 42
2.2 Summary of Contacts 45
2.3 Summary of Markers (Bit) 46
2.4 Summary of Marker Words, Double Words,
Floating Point Words 47
2.5 Data Structures 48
2.5.1 Summary of Defined Data Structures 48
2.5.1.1 New Data Structures 50
2.6 Operands 51
2.7 Operations 52
2.8 System Operands 54
2.8.1 Definition/Use 54
2.8.2 System Markers M1.1 - M4.3 55
2.8.3 System Marker Words, Double words,
Floating Point Words 57
2.9 Notes 60
2.10 BSW Version 6.0, Module/New SFBs 61
2.10.1 Loadable Basic Software (BSW) Module Version 6.0 .. 61
2.10.2 SFBs without Configuration 62

Chapter 3 Operation 63
3.1 Operation with the Mouse 64
3.2 Operation with the Keyboard 65
3.2.1 US/German keyboard 65
3.2.2 Keyboard Sections 65
3.2.3 Line Editor 67
3.2.4 Screen Copy, ”Screensave” 68
3.2.5 Key input memory, ”Learn” 69
3.2.6 Key Macros 70
3.2.7 Auto-Repeat Function 70
3.2.8 Special Keys 71

Chapter 4 Programming . . . 73
4.1 General . . . 74
4.2 Dolog AKF Main Menu . . . 75
4.2.1 Status line . . . 76
4.2.2 Pulldown Menues . . . 76
4.2.3 Help Function . . . 77
4.3 Edit . . . 78
4.3.1 Overview . . . 79
4.3.2 Blocks . . . 83
4.3.3 Symbols and Comments . . . 152
4.3.4 Equipment List . . . 162
4.3.5 Title Block . . . 171
4.3.6 Replace Signals . . . 174
4.3.7 Compile (Blocks) . . . 178
4.3.8 Translate (SYM/COM Blocks) . . . 179
4.3.9 DOLOG-SFB Address List . . . 184
4.3.10 Command File . . . 188
4.3.11 Data Structures . . . 191
4.3.12 Closed-Loop Control . . . 200
4.4 Load . . . 211
4.4.1 Program Link . . . 212
4.4.2 Program to PLC . . . 212
4.4.3 Exchange Online . . . 214
4.4.4 Read Equipment List . . . 217
4.4.5 Read Out PLC . . . 218
4.4.6 Compare . . . 219
4.4.7 Set Date/Time . . . 222
4.4.8 Optimizing of PLC Memory . . . 224
4.4.9 Signal Memory (PLC → PADT) . . . 224
4.4.10 Signal Memory (PADT → PLC) . . . 225
4.4.11 Initial Value to PLC . . . 226
4.4.12 Archive Memory Area . . . 228
4.4.13 Restore Memory Area . . . 230
4.4.14 EPROM handling . . . 232
4.5 Online . . . 236
4.5.1 StArt PLC . . . 236
4.5.2 StOp PLC . . . 237
4.5.3 Dynamic Status Display . . . 237
4.5.4 Status List . . . 251
4.5.5 Control List . . . 252
4.5.6 Force List . . . 254
4.5.7 Terminal Mode . . . 255
4.5.8 Closed Loop Control . . . 256
4.6 Print . . . 263
4.6.1 Overview . . . 263

4.6.2 Program Log 265
4.6.3 Symbols and comments 268
4.6.4 Equipment List 268
4.6.5 Cross-Reference List 269
4.6.6 Signal Occupancy List 271
4.6.7 Station Setup 273
4.6.8 Command File 274
4.6.9 Signal Memory Contents 275
4.6.10 Data Structure 276
4.7 Special 277
4.7.1 Directory 278
4.7.2 Archive the Station 282
4.7.3 Restore a Station 285
4.7.4 Erase a Station 287
4.7.5 Format Disks 287
4.7.6 Copy Disk Complete 289
4.7.7 Erase Files 290
4.7.8 Copy of Files 291
4.7.9 Import 295
4.7.10 Export 305
4.7.11 Operating System MS-DOS 309
4.7.12 System Information 309
4.7.13 End of Station Handling 310
4.8 Setup 311
4.8.1 Plant 312
4.8.2 Station 313
4.8.3 Networking 331
4.8.4 Print 335
4.8.5 Colors 341

Chapter 5 For Users of BSW < 6.0 343
5.1 Loadable Basic Software Version 6.0 344
5.2 Configure Basic Software 344
5.3 Dolog SFB Address List 344
5.4 POS 102/POS 112 with AKF35 and BSW Version 6.0 345
5.5 Intelligent Function Modules (Front) 345
5.5.1 Flowchart 345
5.5.2 First-time Parametrization of PLC 347
5.5.3 Set Up ”RAMZU Expert” with Terminal Mode 348
5.5.3.1 Select Terminal Mode 348
5.5.3.2 Select Memory Area 348
5.5.3.3 Set Up Memory Area 349
5.5.3.4 Enter ”RAMZU Expert” 350
5.6 Connection User Program Dolog B/ Dolog AKF 352

Part IV Formal Operands of SFBs 355

Chapter 1 SFB Formal Operands 357
1.1 General Information 358
1.2 Tables of the Formal Operands 359

Part V Dolog AKF for Beginners 475

Chapter 1 Introduction 477
1.1 General Information 478
1.2 Programming Components 479
1.3 Basic Functions 480

Chapter 2 Structured Programming 481
2.1 General Information 482
2.2 Program Structure 483

Chapter 3 Special Languages of Programming 485
3.1 Instruction List IL 487
3.2 Ladder Diagram LD 489
3.3 Function Block Diagram FBD 490

Chapter 4 Blocks 491
4.1 Block Types 492
4.2 Organization Block OB 493
4.2.1 OB1 493
4.2.2 OBi 495
4.3 Program Block PB 496
4.4 Function Block FB 497
4.5 Standard Function Block SFB 499
4.6 Data Structures 500
4.7 SYM/COM Block 503

Chapter 5 First Programming Steps with AKF35 505
5.1 General Information 506
5.2 Preparatory steps 506
5.3 Problem Statement 507
5.4 Parameters of the Programming Example 508
5.5 Programming 509
5.5.1 Call Program 509

5.5.2 Set Plant/Station 510
5.5.3 Program Presettings 511
5.5.4 Edit Equipment List 512
5.5.4.1 Activate equipment list editor 512
5.5.4.2 Set Subrack 512
5.5.4.3 Enter Module 513
5.5.4.4 Enter/Alter Comments for Module 514
5.5.4.5 Terminate and Store Equipment List 514
5.5.5 Assign Symbols and Comments (SYMCOM block) . . 515
5.5.6 Edit Program (Blocks) 521
5.5.6.1 Open Block Editor 521
5.5.6.2 Edit FB1 521
5.5.6.3 Edit OB1 528
5.5.7 Connect PLC / Networking PLC ↔ PADT 532
5.5.8 First-time Parameter Assignment 533
5.5.9 Link Program 535
5.5.10 Load Basic Software and Program to PLC 535
5.5.11 Start Program 536
5.5.12 Dynamic Status Display 537
5.6 Further Exercise (Solution) 538
5.7 Remarks about the Program Documentation 539
5.8 Remarks about Data Security 539

Chapter 6 Task Description of AKF Station ”Example” 541
6.1 General Information 542
6.2 Problem Definition 543
6.3 Station 1 545
6.4 Station 2 546
6.5 Station 3 548

AKF User Club 551

Part VI Index 553

Index 555

Part VII Appendix 569

Part I
Getting Started

Chapter 1
Getting Started

1.1 General Information

Before you start configuring your A350/A500 user program with this software package, you should become familiar with at least the contents of the following chapters:

in the ”A350 User’s Manual, A91V.12-234678”

- ❐ Chapter 1.3 General Mode of Operation
- ❐ Chapter 1.5 Recommended Peripherals
- ❐ Chapter 1.6 Programming
- ❐ Chapter 3.7 Addressing the Memory
- ❐ Chapter 3.8 Checklist for Initial Start-up and Test

in the ”A500 User Manual, A91M.12-279330”

- ❐ Chapter 1.3 General Mode of Operation
- ❐ Chapter 1.6 Programming
- ❐ Chapter 3.8 Addressing the Memory
- ❐ Chapter 3.9 Checklist for Initial Start-up and Test

The sections mentioned below also can be used as a ”guide” through this documentation and through the PLC configuration.

Before you begin configuration, you should become familiar with the checklists, which point out requirements and sources of documentation (starting on page 6).

Once you have reached the actual configuration of your user program, flowcharts will be of further use. These will guide you step by step from the configuration via program transmission to the start-up of your user program (starting on page 8).

Chapter 2
Checklists / Programming Sequence

This chapter contains information about how to proceed during A350/A500 configuration. It contains checklists for installation, configuration, program transmission/start-up and test. Cross references refer you to additional sources of information. This is followed by a ”flowchart”.

2.1 Checklists

The following chapters point out the steps that must not be forgotten under any circumstances.

Before you begin configuration of your user program with the AKF35 software package for Modicon A350 or Modicon A500, you should

- go through the checklist corresponding to the particular configuration step
- read the details at the corresponding cross-references (UM A350 means ”User Manual A350”).

2.1.1 Installation

- Does your programming panel fulfill the requirements (Part II, page 24)
- Did you make backup copies of the original diskettes (Part II, page 27)
- Did you install the software (Part II, as of page 29)
- Did you call AKF35 (Part II, page 30)

2.1.2 Configuration

- Did you read the chapter ”Equipping and Defining the I/O Nodes” (UM A350 or A500)
- If you are carrying out a COMAKF configuration, did you:
  - carry out the steps required in AKF (as of page 17)
  - configure the communications tables (software package COMAKF)
  - call AKF35 again
- Did you familiarize yourself with the functions of the operating keys and the mouse (as of page 63)

- ❐ Did you select the pulldown menues in the main menu and familiarize yourself with the menu texts and their functions (as of page 73).
  The functions are explained in help texts which you can either call at the PADT (page 77) or read about (part III)

- ❐ If you are a beginner:
  Did you read the comments about structured programming (Part V, as of page 481) and study the configuration example (as of page 541).
- ❐ Otherwise:
  Using the flowchart (as of page 8) you can begin configuration of your user program.

2.1.3 Program Transmission/Start-up/Test

- ❐ Did you prepare the program for the PLC transmission (”Program Link”, as of page 212)

- ❐ Did you prepare the PLC for program transmission:
  - Standardization as of BSW version 5.05 with ”SSN” (A500 BSW Extensions Version 6.0)[2)]
  - Did you read and study the chapter ”Checklist Initial Start-up and Test” (PC* First-time Parameter Assignment, UM A350 or A500)

2.1.4 EPROM Operation

- ❐ Do you want to transmit to EPROM after a positive program test (page 16, as of page 232)

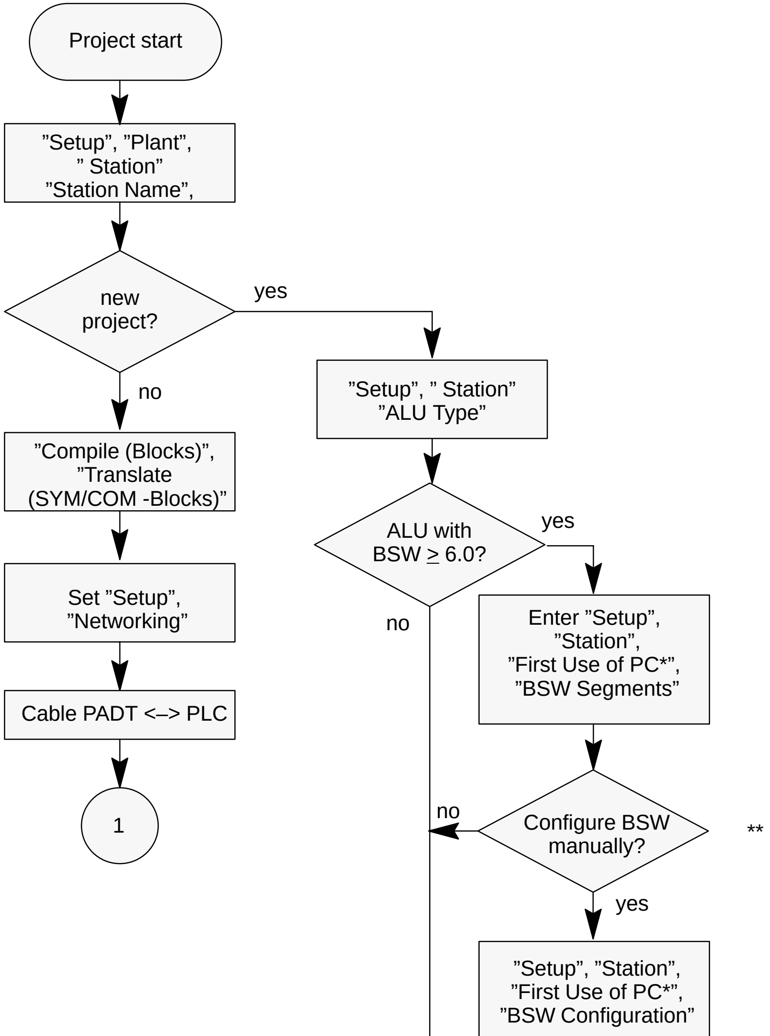
The PLC is now installed for continuous operation and configuration with AKF35 is therefore terminated.

2) in german language

2.2 Programming Sequence

The following pages show you in the form of a program flowchart how to proceed in A350/A500 system processing with AKF35.

☞ **Note** The configuration for the programmable controllers is described in part III. AKF35 instructions for beginners can be found in Part V (with detailed example as of page 505).



** Configure the planned end configuration since BSW modules cannot be exchanged online; then link the BSW after the first use of PC* and transmit them to the PLC, see also as of page 320.

Figure 1 Flowchart System Processing

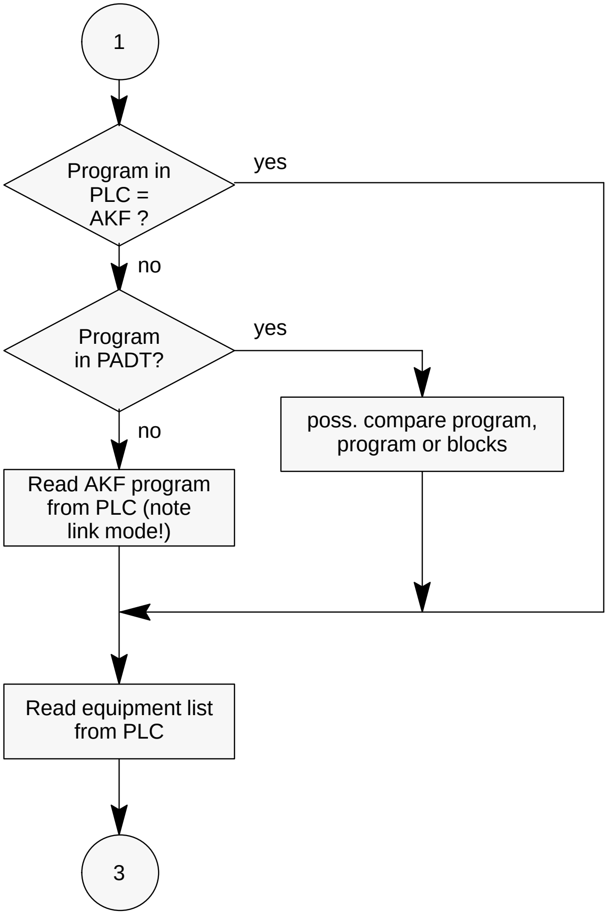


Figure 2 Flowchart System Processing (continued from Figure 1)

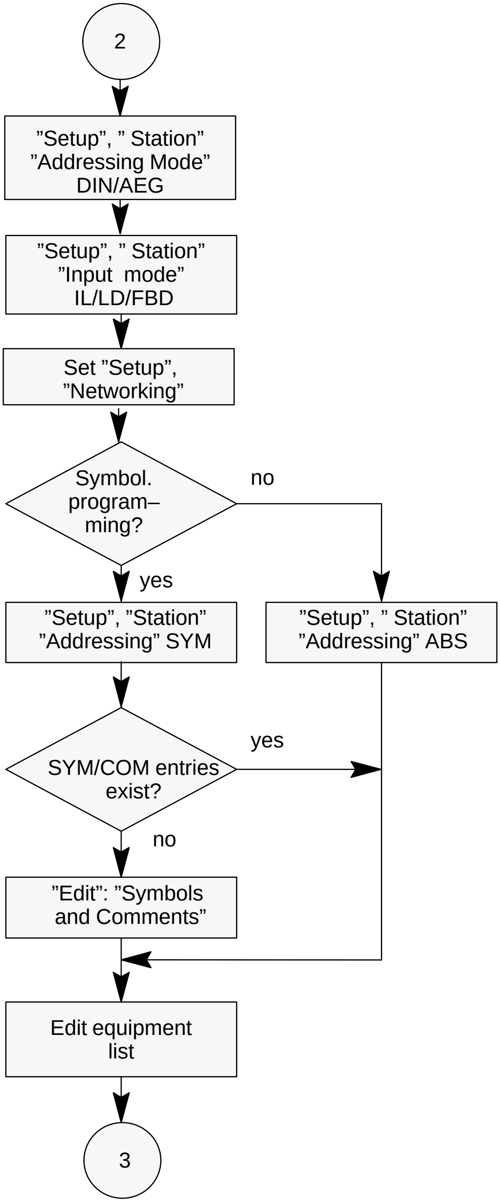


Figure 3 Flowchart System Processing (continued from Figure 2)

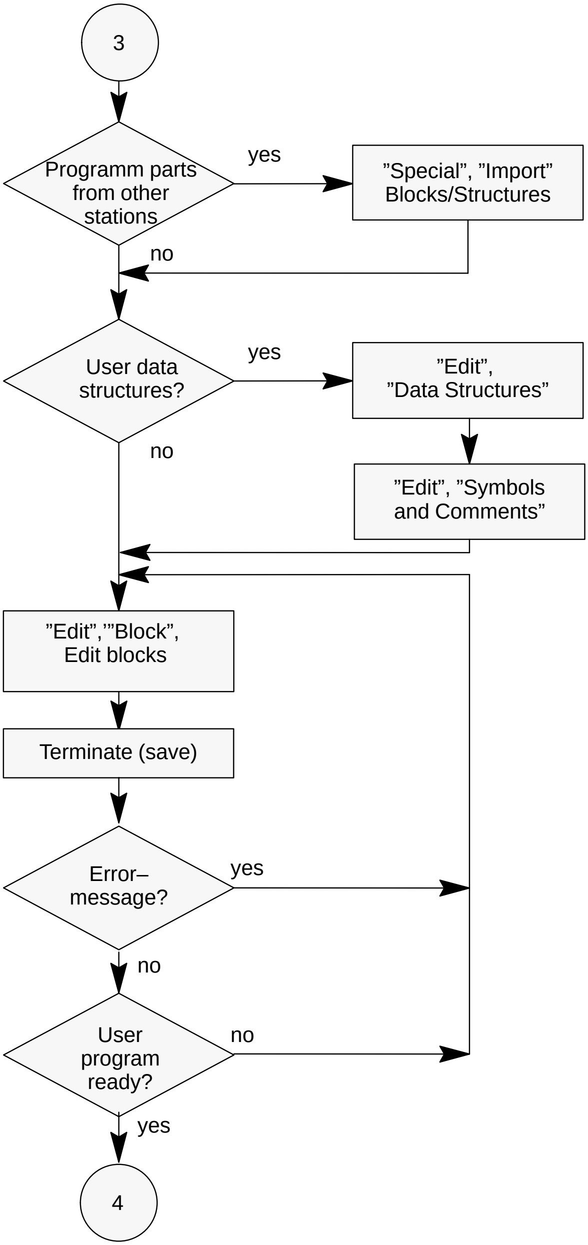


Figure 4 Flowchart System Processing (continued from Figure 3)

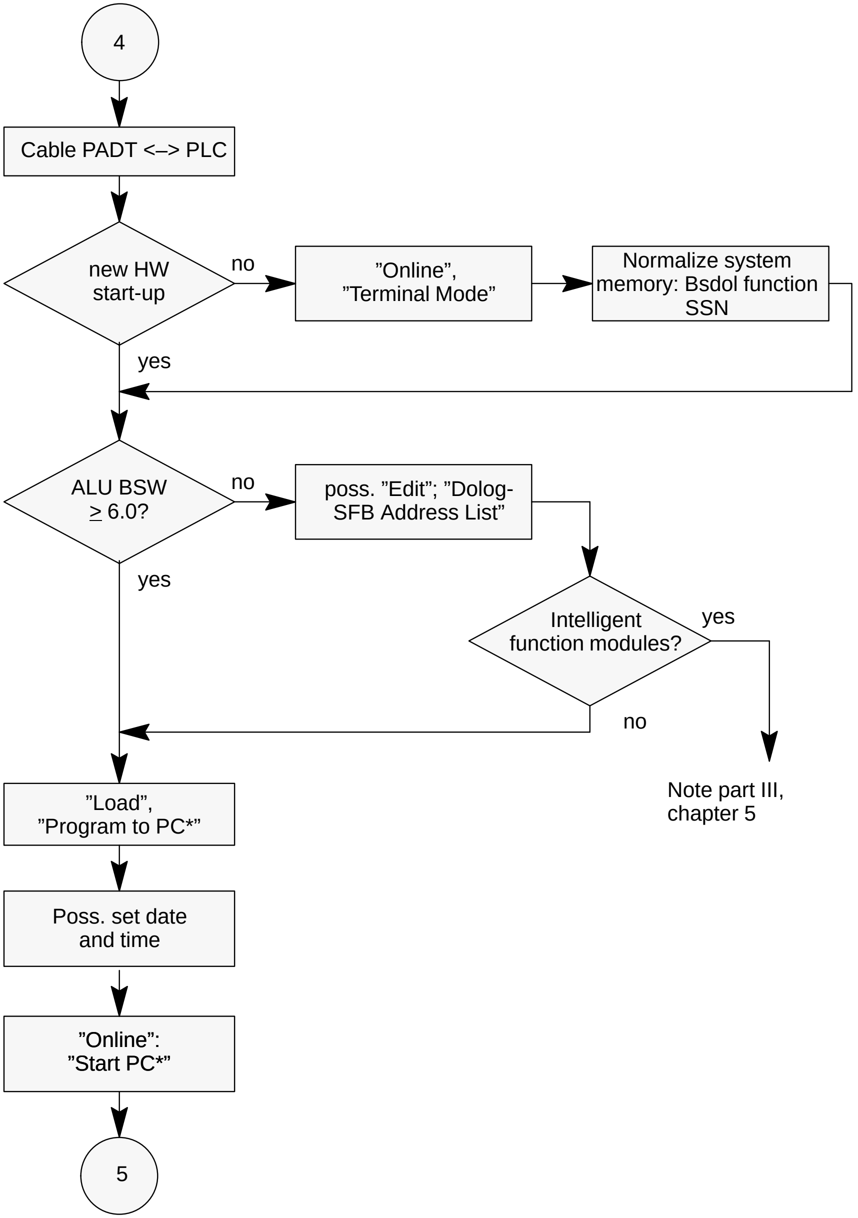


Figure 5 Flowchart System Processing (continued from Figure 4)

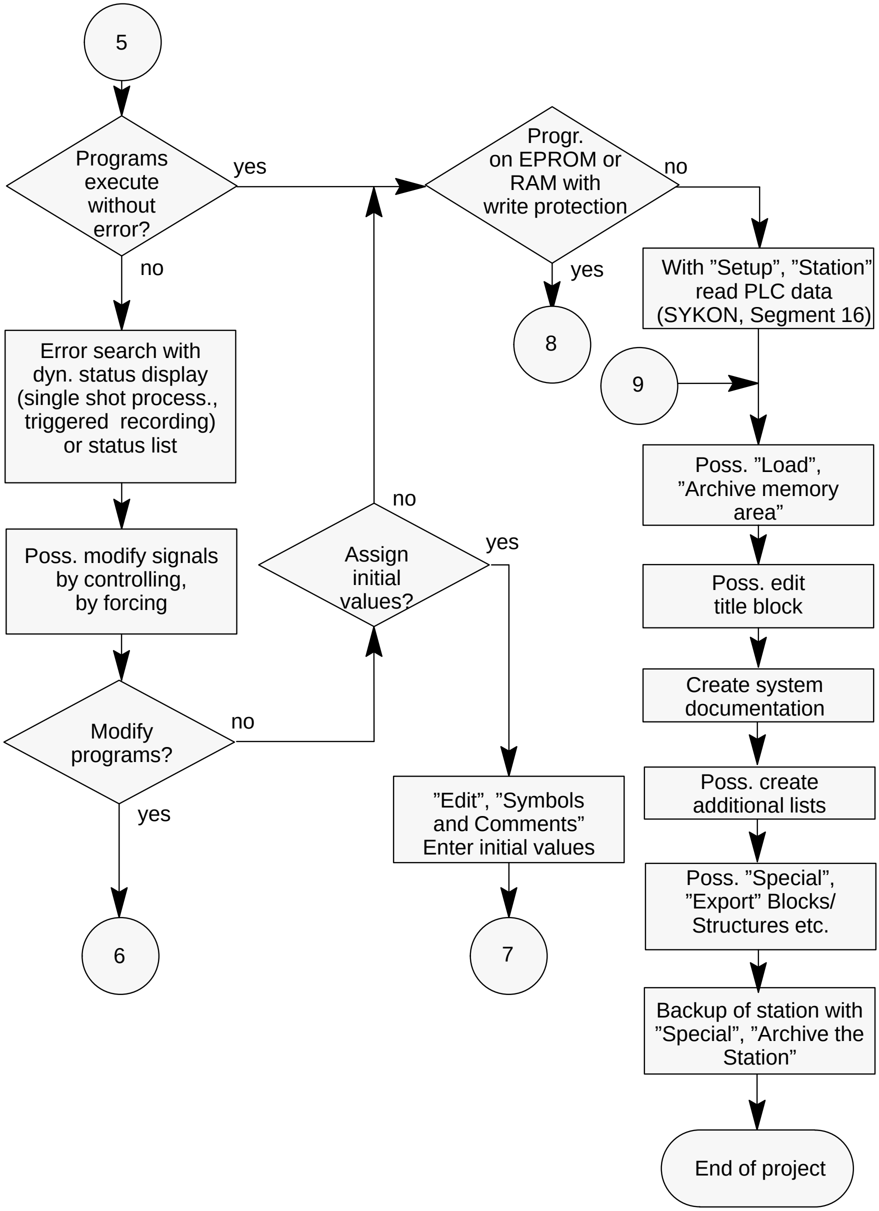


Figure 6 Flowchart System Processing (continued from Figure 5)

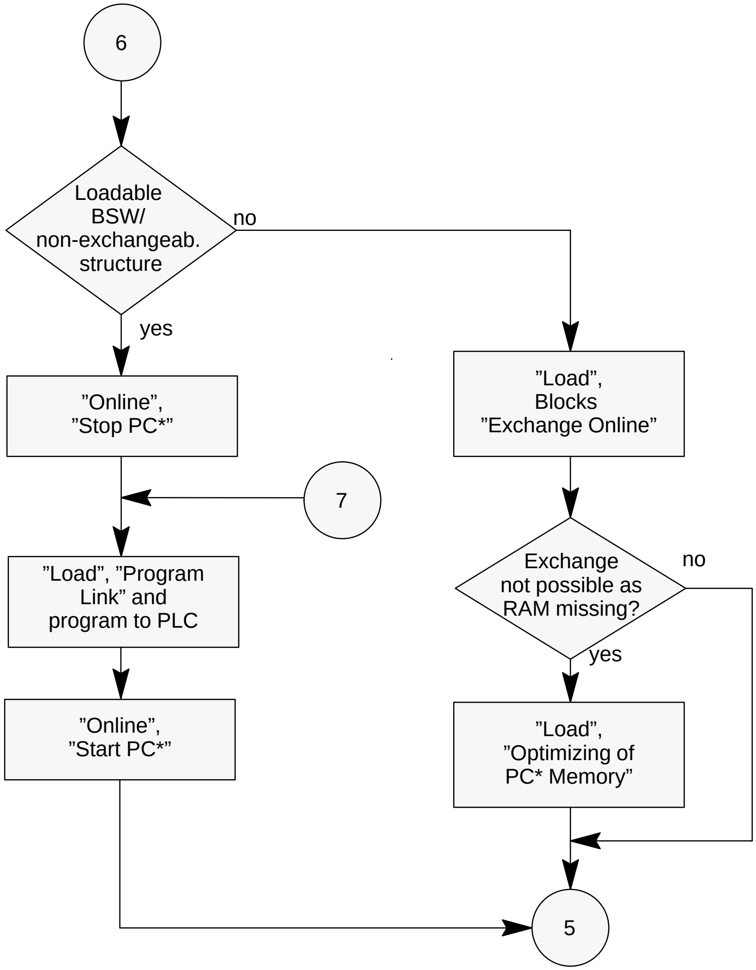


Figure 7 Flowchart System Processing (about Figure 6)

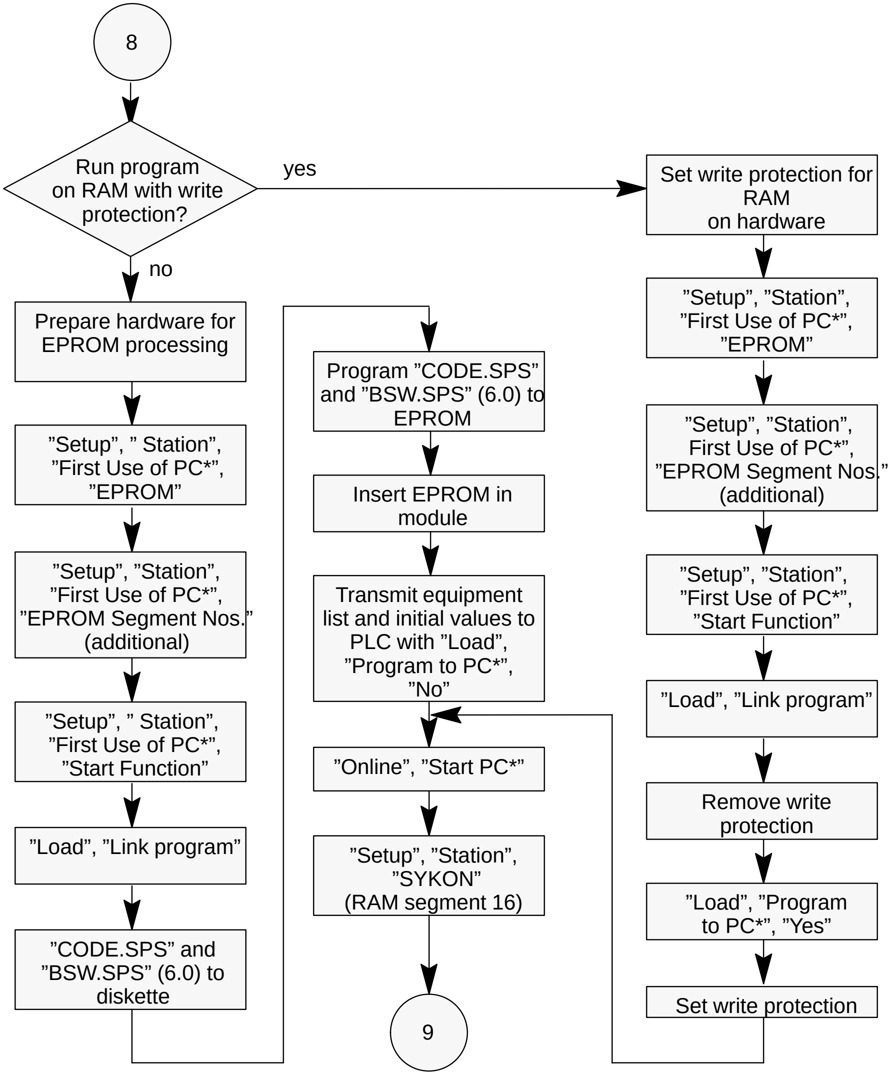
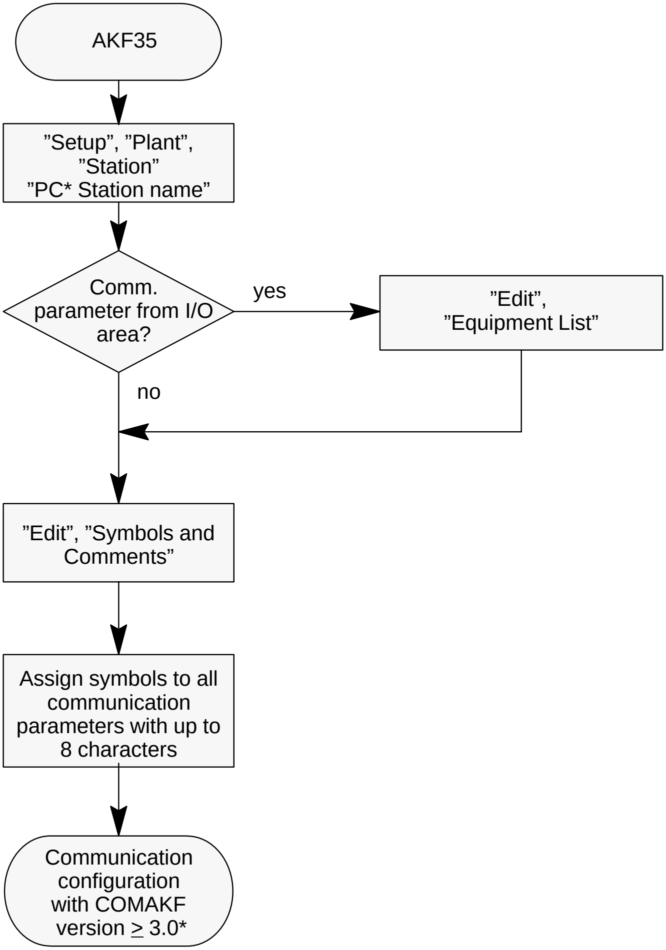


Figure 8 User program to EPROM / Write-Protected RAM

2.3 COMAKF Initialization in AKF

☞ **Note** The steps described here are only necessary if you want to have a communication configuration with COMAKF.

The following page shows you in the form of a program flowchart what must be done in AKF35 before you can begin with the COMAKF configuration.



* After link configuration, continue work with the flowcharts from chapter 2.2.

Figure 9 COMAKF Initialization Flowchart

Part II
Installation Instructions

Chapter 1
General

1.1 Summary

This publication describes the installation of the Modicon software packages. The software is valid for IBM-compatible personal computers which use the MS-DOS operating system, especially for the IBM-compatible AEG programming panels.

The Modicon software is installed using menues. This ensures that user files which already exist are not destroyed. You are recommended to initialize your personal computer for Modicon software with this software, even if you already installed the operating system (for computers with hard disk).

☞ **Note** Programming panels are called PADT (programming and debugging tool) in the following text.

You can program the A350 and A500 programmable controllers with your PADT in connection with the provided software. The programs can be written, documented and archived using the programming panel.
Programming on EPROM is also possible using the EPROM programming panels (EPS 2000, EPS 386).
The MS-DOS operating system also permits the use of the programming panel as a personal computer.

Chapter 2 Installation

This chapter provides information about the programming panel selection, the software installation on the programming panel and the software call

☞ **Note** The features of the software packages can be found in the Modicon catalogues for the programmable controllers.

2.1 Device selection/Requirements

2.1.1 Important Remarks

☞ **Note** The P510-40 and P610 devices as of device index .03 guarantee optimal operation. As of this index, the programming panels are delivered from the factory with installed memory extension (incl. driver installation). The device index can be found on the silver name-plate (lower side of device), marked with Rev.

Caution For MS-DOS versions as of 3.3: The MS-DOS command APPEND may not be used in connection with this software and the APPEND command may not lead to AKF directories.

Caution Reliable processing of the AEG software is not possible with MS-DOS versions other than version 3.x or 4.1 due to insufficient compatibility.

Caution Do not start memory-resident or interrupt-controlled PLC programs parallel to the AEG software (memory requirements too large).

2.1.2 Requirements P510-40/P610

To program a A350 or A500 with the existing software on a P510-40 or P610 you need

- a P510-40/P610 programming panel (PADT) with
  - file CONFIG.SYS: entry for ”Buffers” = 20, ”Files” = 20
  - unused user memory (MS-DOS) before AKF35 call: at least 570 kByte
  - no memory-resident programs started on PADT
  - MS-DOS versions 3.x / 4.1

- Interfaces of the PADT:
  - COM1 for PLC operation on the PADT (for program transmission, initial start-up)
  - COM2 optional, for connection of a mouse or printer (for configuration, documentation, monitoring)
  - LPT1 optional, for connection of a printer (for configuration, documentation, monitoring)

- Recommendation:
  PADT with user memory extension (EMS), for P510-40, P610 < .03

2.1.3 Requirements for IBM-compatible PLC

To program a A350 or A500 with the existing software on an IBM-compatible PLC you need

- an IBM-compatible programming panel (PADT) with
  - hard disk (at least 40 MByte)
  - diskette drive 3 1/2 ” or 5 1/4 ”
  - EGA card
  - file CONFIG.SYS: entry for ”Buffers” = 20, ”Files” = 20
  - 640 kByte RAM, of which unused user memory (MS-DOS) before AKF35 call: at least 570 kByte
  - US keyboard driver
  - no memory-resident programs started on the PADT
  - MS-DOS version 3.x / 4.1

- PADT interfaces:
  - COM1 for PLC operation on the PADT (for program transmission, initial start-up)
  - COM2 optional, for connection of a mouse or printer (for configuration, documentation, monitoring)
  - LPT1 optional, for connection of a printer (for configuration, documentation, monitoring)

- Recommendation:
  PADT with user memory extension (EMS)

2.2 Creating a backup copy

You need a backup copy for each original diskette contained in the software pak-kage. The required number (corresponding to the number of original diskettes) of empty diskettes should be provided for creating such a copy.

You can create the backup copies using the AEG main menu or at the operating system level.

If you are in the AEG main menu, proceed as follows:

Step 1 Select the menu ”User1” with the cursor keys, reference characters or the mouse.

Step 2 Select the function ”Diskcopy a: a:” and execute it with <Return> or execute directly with the reference characters.

Step 3 Follow the directives on the monitor.

Step 4 Answer the question ”Create further copies (Y/N)?_” with ”Y” until all the original diskettes have been copied. Then press ”N” and leave the command menu.

Step 5 There is an immediate switch to the AEG main menu.

You can reach the operating system level with ”Exit”.

If you are in the operating system, proceed as follows:

Step 1 Enter `Diskcopy a: a:` <Return> (or e.g. `b: b:`).

Step 2 Follow the commands on the monitor.

Step 3 Answer the question ”Create further copies (Y/N)?” with ”Y” until all the original diskettes have been copied. Then press ”N” and leave the command menu.

Note Label the backup copy to correspond to the original diskette. The original diskette should now be kept in a safe place so that no data is destroyed.

2.3 Installation of the Software

You can install this software package on the hard disk.

☞ **Note** Please create the backup copy first and do not work on the original, see chapter 2.2.

2.3.1 Installation on the Hard Disk

Step 1 Insert the backup copy in the diskette drive (e.g. A:)

Step 2 Change to the drive in which you want to install (e.g. C:)

Step 3 Answer the question `C:` with `A:INSTAL` and terminate with <Return>.

Step 4 Follow the commands on the monitor.

Step 5 The code for the drive (e.g. C:) appears on the monitor when the software is installed.

Step 6 Remove diskette from the drive.

Step 7 Reset the PADT with <Ctrl>+<Alt>+<Del> (”warm restart”) when the installation has been ended.

Step 8 If necessary, enter the date and time.

Step 9 At the operating system level the code of the user drive appears (e.g. C:).

2.4 Software Call Dolog AKF → A350 / A500

Once you have installed the software on the user drive, you can call it from the hard disk. Start the software at the MS-DOS level (see Table 1, page 30).

Table 1 Software Call

Input	Recommendation for monitor
AKF35<Return>	–
AKF35 /COL<Return>	Color monitor
AFK35 /GR<Return>	Liquid crystal displays and plasma monitor
AKF35 /BW<Return>	Black-and-white monitor
AKF35 /NOSPS<Return>	Valid for mouse operation at COM1 interface

You can now begin writing your program.

Part III
Programming Instructions

Chapter 1
Introduction

1.1 General

This programming guide is a reference manual for using the Dolog AKF → A350/A500 software with version number 6.0.

The software is used for structured programming of PLC user programs with the modern window technique (window, pulldown menu).

Note After installation, you can call a summary of the helptexts by pressing the <F10> key twice. There you can find general information about the Dolog AKF software and the operator interface under ”Information about Help” and ”Dolog AKF Software Overview”.

The order of the texts of the ”Programming” chapter is oriented to the structure of the Dolog AKF main menu from left to right although the programming actually begins with the setup functions. Within the description, the alphabetical index of keywords tells you where in the text which function is described.

1.2 New Features of Versions 5.0 to 6.0

If you have received a new version of the AKF software and installed it on your PADT, please always link each block of the user program which was created with a different AKF version.
Please load the complete linked program into the PLC afterwards.

- **General Information**
  - Data Structures
    Tables of parameters for further structuring of user programs/standard function blocks
  - <Ctrl>+<Reference character>
    Execution of editor functions without selecting the menu (was previously possible with function keys)
  - <Ctrl>+<Return>
    Call of the pulldown menu in the editor
  - <F1>, <F2>
    ”Zoom-in” and ”Zoom-out” in editors and helptexts
  - New standard function blocks in AKF35, version 6.0

only BSW 6.0	GSW $\geq$ 5.05		
ISTD_POS	KPID	VIP+	COS
EINR_POS	PID	VIPS+	TAN
AUTO_POS	PIDP	SA03	ASIN
HAND_POS	PI	SA03I	ACOS
	ZR	SA03E	ATAN
	DR	EX	LG
	PBM	LN	GVERH
	PDM	GAWE4	GMINI
	PT1	SEIG	GMAXI
	PT2	LDSG	GVORL
	DT1	SBVE	GPGON
	PDT1	GRAD1	GBGRZ
	IB	GRAD2	GTZONE
	AB1	GNEG	GSPM
	AB2	GABS	GAEM
	TZ	GQAD1	GRZMH
	STOE	GQAD2	POLY
	O-REG	SIN	POT

  - ALU types ALU 021 and ALU 071 with loadable basic software, i.e. with basic software version 6.0

Edit

Block	Copy network (in correction mode)
Symbols and Comments	only one SYM/COM block
Equipment list	also ”Number”, ”Directory structure” also modules: POS102 POS112 SAI103 SAA103 OIS–I
Compile (Blocks)	Transmit OB, PB, FB to data base
Translate (SYM/COM blocks)	DB0 ... DB9 converted to a SYM/COM block
Data Structures	is used to create data structures
Closed-loop controll	runtime system used for closed-loop control with AKF35

Load

Compare	Now one menu option under ”Load”
Set date/time	Transfer from PADT to PLC
Archive memory area	Transfer memory area from PLC to PADT (e.g. for TESY)
Restore memory area	Memory area from PADT to PLC

Online

Terminal Mode	Leave the function with <F9>
Closed-loop control	runtime system used for closed-loop control with AKF35

Print

Generally for print	Page mode and start page number
Data Structure	

- **Special**
  - Directory ,
    Erase Files,
    Copy Files — Selection of "AKF blocks" and "DOS files"
  - Import — Import of data structures, blocks and symbols and comments
  - Export — Export of data structures, blocks and symbols and comments
  - System Informations — previously "unused memory"
- **Setup**
  - Station — ALU type as selection window,
    new ALUs with BSW version 6.0:
    ALU 021, ALU 071;
    Configure basic software for ALU 021, ALU 071

1.3 Basic Structure of the Software (V6.0)

The software is divided into the following rough structure:

Edit

- Overview
- Block
- Symbols and Comments
- Equipment List
- Title Block
- Replace Signals
- Compile (Blocks)
- Translate (SYM/COM-Blocks)
- Dolog-SFB Address List
- Command File
- Data Structures
- Closed-loop control

Load

- Program Link
- Program to PC*
- Exchange Online
- Read Equipment List
- Read out PC*
- Compare
- Set Date/Time
- Optimizing of PC* memory
- Signal Memory (PC* → PADT)
- Signal Memory (PADT → PC*)
- Initial Value to PC*
- Archive memory area
- Restore memory area
- EPROM-Handling

Online

- Start PC*
- Stop PC*
- Dynamic status display
- Status list
- Control list
- Force list
- Terminal Mode
- Closed-loop control

Print

- Overview
- Program log
- Symbols and comments
- Equipment list
- Cross-reference list
- Signal occupancy list
- Station Setup
- Command file
- Signal memory contents
- Data structure

Special

- Directory
- Archive the station
- Restore a station
- Erase a station
- Format disks
- Copy disk complete
- Erase files
- Copy files
- Import
- Export
- Operating system MS-DOS
- System informations
- End of station handling

Setup

- System
- Station
- Networking
- Print
- Colors

Chapter 2
Validity Scopes and System Operands

The following chapter contains the most important system data for configuration with Dolog AKF → A350/A500.

2.1 Validity Scopes of the Software Dolog AKF → A350/A500

- One organization block OB1 and one close-loop control organization block, OB2 to OB999,
  999 program blocks, PB1 to PB999,
  999 user function blocks, FB1 to FB999 and
  999 standard function blocks SFB1 to SFB999 are possible.

- A maximum of 999 networks is possible per block in organization blocks OBs, in the program blocks (PBs) and in the function blocks (FBs). The number of possible networks depends on the complexity of the individual instructions. If no comments, labels or parameters were used, it is limited by the maximum number of instructions per block (approx. 2000).

- In program block networks a maximum of 16 outputs is possible, but all with the same potential.

- A user function block may have a maximum of 45 input or output parameters, but together no more than 64 parameters. A data structure counts here as a ”parameter”.

- In ladder diagrams and function block diagrams a maximum of 16 outputs is possible, but all with the same potential.

- Program and function block calls in LD/FBD may only stand alone in a network.

- Networks may have a maximum length of 255 IL lines.

- The maximum nesting depth for bracketed expressions is 13, i.e. as many as 13 brackets may be open at the same time.

- A maximum of 24 labels per network is possible within a user function block.

- Standard function blocks (SFB) may also be called from a user function block (FB).

- A program block may be called a number of times. A recursive call is also possible, but the nesting depth may not exceed 10. In a recursive call, a program block may thus only be called conditionally.

Caution In order to keep your program executable, you must configurate an abort conditon within 10 calls when nesting in the program block.

- The signal comment length is limited to a total of 48 characters. 8 characters are for the symbol names and a maximum of 40 for comments.

- There are no jumps, only block calls, in the ladder diagram/function block diagram. Processing of the called block is followed by a return to the calling point.
  In the ladder diagram, 16 signals can theoretically be linked in parallel and 7 signals plus output serially.

- In the function block diagram, a maximum of 6 discrete elements next to one another horizontally and, for example, 12 elements with 2 inputs one under the other is possible.

- Outputs may not be negated.

- A maximum of 18 signals may be set (except system operands and actual time value).

- A maximum of 18 signals may be forced (except system operands and the actual time value).

- Up to 64 control loops may be configured. The basic sampling time is 0.01 to 1.27 seconds.

- The markers (bits), marker words, marker double words and marker floating point words may only have the following values:

Table 2 Operand Value Ranges

Parameter	Size	Value range
generally		
marker (bit):	1 bit	0 / 1
marker words:	16 bit	-32 768 to +32 767
marker double words:	32 bit	-2 147 483 648 to +2 147 483 647
marker floating point words:	32 bit	± (1.1755E-38 to 3.60282E38) sign +8 bit exponent + 23 bit mantissa
for structures of FBs and SFBs:		
marker	1 byte	0 / 1
byte	1 byte	-256 to +255
word	2 byte	-32 768 to +32 767
double word	4 byte	-2 147 483 648 to +2 147 483 647
floating point word	4 byte	± (1.1755E-38 to 3.60282E38) sign +8 bit exponent + 23 bit mantissa
pointer	4 byte	segment and offset (internal)
stream	variable	freely defined by system (internal)

2.2 Summary of Contacts

The designation of the contacts depends on the hardware module used. The address must be preceded by the slot address of the module in the subrack.

Table 3 Summary of Contacts

DEP 0xx DAP 0xx	DEP 1x2 DAP 1x2	DEA H1 DEA K1 Input	Outp.	DAP 102 103	DAP 104	DAP 106	DIN
A02	A32	---	A02	A32	A32	A32	.1
A04	A30	---	A04	A30	A30	A30	.2
A06	A28	---	A06	A28	A28	A28	.3
A08	A26	---	A08	A26	A26	A26	.4
A10	A24	---	A10	A24	A24	A24	.5
A12	A22	---	A12	A22	A22	A22	.6
A14	A20	---	A14	A20	A20	A20	.7
A16	A18	---	A16	A18	A18	A18	.8
A18	A16	A18	A18	A16	---	A16	.9
A20	A14	A20	A20	A14	---	A14	.10
A22	A12	A22	A22	A12	---	A12	.11
A24	A10	A24	A24	A10	---	A10	.12
A26	A08	A26	A26	A08	---	A08	.13
A28	A06	A28	A28	A06	---	A06	.14
A30	A04	A30	A30	A04	---	A04	.15
A32	A02	A32	A32	A02	---	A02	.16
E02	E32	E02	---	E32	E32	---	.17
E04	E30	E04	---	E30	E30	---	.18
E06	E28	E06	---	E28	E28	---	.19
E08	E26	E08	---	E26	E26	---	.20
E10	E24	E10	---	E24	E24	---	.21
E12	E22	E12	---	E22	E22	---	.22
E14	E20	E14	---	E20	E20	---	.23
E16	E18	E16	---	E18	E18	---	.24
E18	E16	E18	---	E16	---	---	.25
E20	E14	E20	---	E14	---	---	.26
E22	E12	E22	---	E12	---	---	.27
E24	E10	E24	---	E10	---	---	.28
E26	E08	E26	---	E08	---	---	.29
E28	E06	E28	---	E06	---	---	.30
E30	E04	E30	---	E04	---	---	.31
E32	E02	E32	---	E02	---	---	.32

2.3 Summary of Markers (Bit)

Marker (own address area with 10000 bit)

Table 4 Summary of the available markers (DIN– / AEG–addresses)

AEG	DIN	Reservation
1	1.1	System marker
:	:	:
32	1.32	:
33	2.1	:
:	:	:
64	2.32	:
65	3.1	:
:	:	:
96	3.32	:
95	4.1	:
:	:	:
99	4.3	:
100	4.4	Free for user
127	4.32	:
:	:	:
159	5.32	:
:	:	:
191	6.32	:
:	:	:
223	7.32	:
:	:	:
255	8.32	:
:	:	:
799	25.31	:
800	25.32	Reserved for B100 standard images (without B100 = free)
:	:	
3999	125.31	
4000	125.32	Free for user
:	:	:
7999	250.31	:
8000	250.32	Reserved for B500 PV-no. (without B500 = free)
:	:	
9699	304.3	:
9700	304.4	System messages
:	:	:
10000	313.16	System messages

2.4 Summary of Marker Words, Double Words, Floating Point Words

Marker words, marker double words and marker floating point words are displayed in both AEG and DIN as decimal addresses (e.g. MW 100, MD 4000, MF 6000). They are all stored in the address area of max. 10000 words.

Table 5 Summary of the available markers (MW, MD, MF)

AEG	DIN	Reservation	Value range
MW1	MW1	System marker words	from -32 768
:	:	:	to +32 767
MW99	MW99	:	:
MW100	MW100	For timers, counters and comparators	from 0
:	:	of user	to +4 095
MW799	MW799	:	:
MW800	MW800	For B100 standard images	from -32 768
:	:	reserved (without B100 = free)	to +32 767
MW3999	MW3999	:	:
MW4000	MW4000	Free for user	from -32 768
:	:	:	to +32 767
MW7999	MW7999	:	:
MW8000	MW8000	Reserved for B500-PV-no.	from -32 768
:	:	(without B500 = free)	to +32 767
MW10000	MW10000	:	:
MD100	MD100 *)		from -2 147 483 648
:	: :		to +2 147 483 647
MD9999	MD9999 :		:
MF100	MF100 *)		from ±1.1755 E-38
:	: :		to ±3.40282 E+38
MF9999	MF9999 :		:

*) Marker words, double words and floating point words occupy the same physical memory space. To avoid any unintentional dual use, you should make a clear distinction between these three marker types in the range from MW/MF/MD 4000 to 7999 (e.g. MW: 6000 - 7999; MD: 4000 - 4998; MF: 5000 - 5998). Marker double words and marker floating point words occupy two consecutive addresses and should therefore always begin with an even-numbered address.

2.5 Data Structures

2.5.1 Summary of Defined Data Structures

This list defines the structures provided with the software and where they are used. In contrast to the user's own structure types, the structures defined below cannot be modified.

☞ **Note** The data structures of the data structure editor not listed here are only used for internal processing.

Table 6 List of the Defined Structure Types

Type	Samples	Use
APOA	64	POS102/POS112, AUTO_POS
AXPA	64	POS102/POS112, ISTD_POS
BIDP	30	Closed-loop control, PIDP
BKID	30	Closed-loop control, KPID
BPID	99	Closed-loop control, PID
BZR	100	Closed-loop control, ZR, DR
EPOA	64	POS102/POS112, EINR_POS
FB	999	Function block
HPOA	64	POS102/POS112, HAND_POS
I	160*	Input parameter
M	10000*	Marker
MD	9999*	Marker double word
MF	9999*	Marker floating point word
MKID	30	Closed-loop control, KPID
MPID	227	Closed-loop control, PID, PIDP, PI
MPOA	64	POS102/POS112, AUTO_POS
MW	10000*	Marker word
OB	999	Organization block
PAB	50	Closed-loop control, AB1

* DIN/AEG: Q160.32/Q160E32, I160.32/I160E32, M313.16/M10000, MW: 10000, MD and MF: always 9999

Type	Samples	Use
PABB	50	Control engineering, AB2
PB	999	Program block
PDR	50	Closed-loop control, DR
PDT	50	Closed-loop control, DT1
PIB	50	Closed-loop control, IB
PIDP	30	Closed-loop control, PIDP
PKID	30	Closed-loop control, KPID
PPBM	99	Closed-loop control, PBM
PPDM	50	Closed-loop control, PDM
PPDT	50	Closed-loop control, PDT1
PPI	99	Closed-loop control, PI
PPID	99	Closed-loop control, PID
PPT	99	Closed-loop control, PT1
PPTT	50	Closed-loop control, PT2
PSTO	50	Closed-loop control, STOE
PZR	50	Closed-loop control, ZR
Q	160*	Output parameter
VBPA	64	POS102/POS112, ISTD_POS
VIA	650	Internal organization information, PI, PBM, PDM, PT1, PT2, DT1, PDT1, IB, AB1, AB2
VIB	160	Internal organization information, KPID, PID, PIDP
VIC	100	Internal organization information, ZR, DR
VIPO	99	POS102/POS112, internal organization information
VRG	500	VIP allocation (internal)
VTZ	50	Closed-loop control, TZ
VVI	99	VIP 101/VIPIPC, VIP+, VIPS+
VVZ	99	VIP 101/VIPIPC, VIP+, VIPS+
ZVT	1	Time management table closed-loop control, O-REG

* DIN/AEG: Q160.32/Q160E32, I160.32/I160E32, M313.16/M10000, MW: 10000, MD and MF: always 9999

2.5.1.1 New Data Structures

The data structure names may contain 4 to 6 letters. The following are possible:

- Names with four letters — example: AAAA
  - sample 1 digit — AAAA9
    max. elements 3 digits — 999
  - sample 2 digits — AAAA99
    max. elements 2 digits — 99
  - sample 3 digits — AAAA999
    max. elements 1 digit — 9

- Names with five letters — example: AAAAA
  - sample 1 digit — AAAAA9
    max. elements 2 digits — 99
  - sample 2 digits — AAAAA99
    max. elements 1 digit — 9

- Names with six letters — example: AAAAAA
  - sample 1 digit — AAAAAA9
    max. elements 1 digit — 9

2.6 Operands

Table 7 Operand Table

Operand-code	Meaning	Operand Parameter range	Operand value range binary	decimal
Q	binary output	*.1 ... *.32	1 bit	0, 1
I	binary input	*.1 ... *.32	1 bit	0, 1
V (decimal)	constant word		max. 16 bit	-32 768 ... +32 767
VH (hexadecimal)			0 ... FFFF	
VO (octal)			0 ... 177777	
VF	floating point word		$\pm$1.1755 E-38 to $\pm$3.40282 E+38	
M	marker (bit)	1.1 ... 313.16	1 bit	0, 1
MW	marker word	1 ... 10 000 #		
MD	marker double word	1 ... 10 000 #	see chapter 2.4	
MF**	marker float. point word	1 ... 10 000 #		
Data structures outside of signal memory (bit, byte, word, double word, floating point word, ”pointer”, ”stream”)				
=	Code for formal operand+			

* Address of the module in the subrack
** An arithmetic coprocessor is required on the ALU when using MF
\# Theoretically possible; please note chapter 2.4.
\+ These operands are specified more precisely for function blocks FB and SFB
in a) Formal operand
b) Actual operand.

The actual operand is the operand assigned at runtime by the program (parameter in FB or SFB call) and may originate in the above table.
The formal operand is the operand which stands as a placeholder within a block (only FB) for an operand of the above table.

2.7 Operations

Table 8 Operations (IL instructions for network creation)

Operation	Operand type OB1, PB	Operand type FB	Explanations
A	I,Q,M	I,Q,M,=	AND (input)
A(	none	none	AND of complex expression (input)
AN	I,Q,M	I,Q,M,=	AND (negated input)
AN(	none	none	AND of complex expression (neg. input)
O	I,Q,M	I,Q,M,=	OR (input)
O(	none	none	OR of complex expression (input)
ON	I,Q,M	I,Q,M,=	OR (negated input)
ON(	none	none	OR of complex expression (neg. input)
)	none	none	”Close bracket” terminates a complex expression
S	I,Q,M,	I,Q,M,=	Set input of memories and counters
R	I,Q,M,	I,Q,M,=	Reset input of memories, counters and timers
LD	MW,MD,MF V,VH,VO,VF	MW,MD,MF= V,VH,VO,VF,=	Load word or word constant (V = binary constant (0/1), DEC), (VH = HEX constant) (VO = Octal constant) (VF = Floating point constant)
LB	-	I,Q,M,=	Load 16 bit from given address into register
T	MW,MD,MF	MW,MD,MF,=	Transfer word (word assignment)
TB	-	Q,M,=	Transfer register to 16 bit from given address
CTU	MWV	MW,V,=	Counter up
CTD	MWV	MW,V=	Counter down
TP	MWV	MW,V,=	Set timer word (pulse)
TEP	MWV	MW,V,=	Set timer word (extended pulse)
TON	MWV	MW,V=	Set timer word (switch-on delay)
TS	MWV	MW,V=	Set timer word (stored switch-on delay)
TOF	MWV	MW,V=	Set timer word (switch-off delay)
NE	MW,MD,MF V,VH,VO,VF	MW,MD,MF,= V,VH,VO,VF,=	Comparison for unequal
EQ	MW,MD,MF V,VH,VO,VF	MW,MD,MF,= V,VH,VO,VF,=	Comparison for equal
GT	MW,MD,MF V,VH,VO,VF	MW,MD,MF,= V,VH,VO,VF,=	Comparison for greater than

Operation	Operand type OB1, PB	Operand type FB	Explanations
LT	MW,MD,MF V,VH,VO,VF	MW,MD,MF,= V,VH,VO,VF,=	Comparison for less than
LE	MW,MD,MF V,VH,VO,VF	MW,MD,MF,= V,VH,VO,VF,=	Comparison for less than or equal
GE	MW,MD,MF V,VH,VO,VF	MW,MD,MF,= V,VH,VO,VF,=	Comparison for greater than or equal
ADD	MW,MD,MF	MW,MD,MF,=	Word addition
SUB	MW,MD,MF	MW,MD,MF,=	Word subtraction
MUL	MW,MD,MF	MW,MD,MF,=	Word multiplication
DIV	MW,MD,MF	MW,MD,MF,=	Word division
NOP	none	none	No operation
=	Q,M	Q,M,=	Assigns an output (bit output) to a preceding OR, AND or memory block
***	none	none	End of network
BE	none	none	End of block, must be at end of each function block. In OB and PBs alone in a network
BC	Name	Name	Block call
BCC	Name	Name	Conditional block call
JI	-	= Jump label	Jump to a label (internal to function block) within a network, an open log. operation, can be executed after the jump
JT	-	= Jump-label	Conditional jump to a label (FB-internal) within a network, JT is instead of assignment of result of operation. Executed for logical ”1”
Jump label:		instruction	Any jump address

2.8 System Operands

2.8.1 Definition/Use

The markers M1.1 to M4.3 and marker words MW1 to MW99 belong to this group. These markers are used by the programmable controller for certain purposes and permit the user to recognize the status of the programmable controller by software.
System operands are bit, word and double word information checked by the system. They are used to analyze a malfunction and to evaluate the system data in the instruction list.
System operands can only be read and used in logical operations. Assignments cannot be made to system operands.

Signal status of the system markers (bit)
If the status is not satisfied (e.g. normal operation), the system markers output the 0 signal.
If the status is satisfied (e.g. malfunction), the system markers output the 1 signal.

Caution **The system markers for the A350 which are different from the A500 are listed twice and marked with ”for A350”.**

2.8.2 System Markers M1.1 - M4.3

Table 9 System Markers M1.1 - M4.3

Name	Address AEG	DIN	"Comments from SYMCOM block" and Explanations	
RESET	M1	M1.1	Starting signal (for 1st scan =1)	
PULSE_1	M2	M1.2	0.3125 Hz blinking rate	
PULSE_2	M3	M1.3	0.625 Hz blinking rate	
PULSE_3	M4	M1.4	1.25 Hz blinking rate	
PULSE_4	M5	M1.5	2.5 Hz blinking rate	
PULSE_5	M6	M1.6	5.0 Hz blinking rate	
ENC_BIT	M7	M1.7	Basic setting encoder monitoring	
STARTUP	M8	M1.8	Startup response of user program (0=restart,1=initial start)	
FIXED"0"	M10	M1.10	Fixed Valence 0	
FIXED"1"	M11	M1.11	Fixed Valence 1	
UKA_BIT1	M12	M1.12	Jumper G (UKA024,ALU0nn,SCU150)	
UKA_BIT2	M13	M1.13	Jumper E (UKA024,ALU0nn)	
VRUN	M14	M1.14	User program running	
SYSLD_1	M18	M1.18	1= No valid data in SYKON segm.	
SYSLD_2	M19	M1.19	1= Automatic SYRES occurred	
I_LOOP	M20	M1.20	Control loop interrupted	(initial start)
I_IOB_TE	M21	M1.21	PEAB timer error	(initial start)
I_OVTEMP	M22	M1.22	Temperature too high	(initial start)
I_BA_UVO	M23	M1.23	Battery undervoltage	(initial start)
I_SIO_PA	M25	M1.25	SIO parity	(initial start)
	M26	M1.26	For ALU150:UKA024 defect or missing	
I_B_COL	M27	M1.27	Collective error DOLOG blocks	(initial start)
I_BATT	M28	M1.28	Battery undervoltage	(initial start)
R_CLOOP	M30	M1.30	Control loop interrupted	(restart)
R_IOB_TE	M31	M1.31	PEAB timer error	(restart)
R_OVTEMP	M32	M1.32	Temperature too high	(restart)
R_BA_UVO	M33	M2.1	Battery undervoltge	(restart)
IOB_TE_U	M36	M2.4	ALU150:UKA024 defect/missing	(restart)
R_B_COL	M37	M2.5	Collective error DOLOG blocks	(restart)
R_BATT	M38	M2.6	Battery undervoltage	(restart)
TRBU_FUL	M40	M2.8	Networking: transfer buffer full	
FORM_ERR	M41	M2.9	Networking: message format error	
SEND_INI	M42	M2.10	Networking: general send init. of all LNN	
SEND_INH	M43	M2.11	Networking: Send inhibit	
RECV_INH	M44	M2.12	Networking: Receive inhibit	
REBU_FUL	M45	M2.13	Networking: Receive buffer full	
KOS_ERR	M46	M2.14	Networking: KOS coll. error	
CODE1	M47	M2.15	Networking: Transparent mode active	
CODE2	M48	M2.16	Networking: Recv.s. if stand. user pr. ignor.	

Name	Address AEG	DIN	"Comments from SYMCOM block" and Explanations
TRTE_REJ	M49	M2.17	Networking: Transfer telegrams rejected
	M50	M2.18	Assigned for B200
PEAB_FLR	M58	M2.26	PEAB power supply failed
UPRG_RUN	M59	M2.27	Reaction to M58/M2.26 (1=User pr. Stop)
UKA_SAF	M60	M2.28	Dropout pilot relay (user-defined)
IOB_CE	M61	M2.29	PEAB coll. error
MEB_CE	M62	M2.30	Memory bus coll. error
TESTRUN	M63	M2.31	Memory read test running
TESTERR	M64	M2.32	Memory read test error
BB_CERR	M65	M3.1	Collective error Modnet 1/SFB

2.8.3 System Marker Words, Double words, Floating Point Words

In this group, the marker words 1 to 99 count as system variables.
These markers are used by the programmable controller for certain purposes and permit the user to recognize statuses of the programmable controller in the software, e.g. date, time, I/O error, etc.
The user cannot assign values to these marker words for his own purposes. The contents of the marker words can only be read out. Their meanings are given below.
Marker words, double words and floating point words occupy the same physical memory space. To avoid any unintentional dual use, the user should make a clear distinction between these three marker types in the range from MW/MD/MF 4000 to MW/MD/MF 7999 (e.g. MW: 6000 - 7999, MD: 4000 - 4998 and MF: 5000 - 5998).
Marker double words and floating point words occupy two consecutive addresses and should therefore always start with an even-numbered address (e.g. MD4254 instead of MD4255).

- Value range for MW from -32,768 to +32,767
- Value range for MD from -2,147,483,648 to +2,147,483,647
- Value range for MF from -1.1755 E-38 to +3.40282 E+38

Table 10 System Marker Words MW1 - MW99

Name	Address	”Comments from SYMCOM Block” and Explanations	
I_10MS	MW2	Max. number 10msec–interrupts/scan	(initial start)
I_SCAN	MW3	Min. number user prog. scans/sec	(initial start)
R_10MS	MW4	Max. number 10msec–interrupts/scan	(restart)
R_SCAN	MW5	Min. number user prog. scans/sec	(restart)
L_10MS	MW6	Number 10msec–interrupts/scan act. value	
L_CYCLE	MW7	Number of user prog. scans/sec act. value	
	MW8	internal	
IOB_TE_A	MW9	PEAB–timing error addr. entry: A: Value = slot address E: Value = slot Address + 1000	
SENSORB_A	MW10	Encoder bit address (user-defined)	
E_TR1_S	MW12		
E_TR1_A	MW13		
E_TR2_S	MW14		
E_TR2_A	MW15		
E_TR3_S	MW16		
E_TR3_A	MW17		
E_TR4_S	MW18		
E_TR4_A	MW19		
E_TR5_S	MW20		
E_TR5_A	MW21		
E_TR6_S	MW22		
E_TR6_A	MW23		
E_TR7_S	MW24		
E_TR7_A	MW25		
A_TR1_S	MW26		
A_TR1_A	MW27		
A_TR2_S	MW28		
A_TR2_A	MW29		
A_TR3_S	MW30		
A_TR3_A	MW31		
A_TR4_S	MW32		
A_TR4_A	MW33		
A_TR5_S	MW34		
A_TR5_A	MW35		
A_TR6_S	MW36		
A_TR6_A	MW37		
A_TR7_S	MW38		
A_TR7_A	MW39		
BS_F01_S	MW40		
BS_F01_A	MW41		
BS_F02_S	MW42		
BS_F02_A	MW43		
BS_F03_S	MW44		
BS_F03_A	MW45		
BS_F04_S	MW46		

Name	Address	”Comments from SYMCOM Block” and Explanations
BS_F04_A	MW47	
BS_F05_S	MW48	
BS_F05_A	MW49	
BS_F06_S	MW50	
BS_F06_A	MW51	
BS_F07_S	MW52	
BS_F07_A	MW53	
BS_F08_S	MW54	
BS_F08_A	MW55	
BS_F09_S	MW56	
BS_F09_A	MW57	
BS_F10_S	MW58	
BS_F10_A	MW59	
YEAR	MW60	Year
MONTH	MW61	Month
DAY	MW62	Day
HOUR	MW63	Hour
MINUTE	MW64	Minute
10TH_SEC	MW65	Tenth second of a minute
SP_SP	MW66	Short-term/permanent storage
BB_A	MW67	Modnet 1/SFB error port address definition
BB_E	MW68	Modnet 1/SFB error number
BAUDRATE	MW71	SW setting of baud rate
V_0	MW72	Constant 0

Table 11 System Marker Double Words MD1 - MD99

Name	Address	”Comments from SYMCOM Block” and Explanations
DC_0	MD74	Contents = ”0”

Table 12 System Marker Floating Point Words MF1 - MF99

Name	Address	”Comments from SYMCOM Block” and Explantions
SW_OFF_ADDR	MF77	
F_0L	MF80	Contents = ”0”
F_1L	MF82	Contents = ”1”
F_10L	MF84	Contents = ”10”
F_50L	MF86	Contents = ”50”
F_90L	MF88	Contents = ”90”

2.9 Notes

The timing elements have a constant time base of 100 msec. The timing elements process 4000 count increments internally, so maximum times of 400 sec (4000 x 100 msec) can be obtained. Longer times can be obtained by connecting a counting element serially. The timing elements are so-called software timing elements, i.e. it is not sufficient to process these just once in order to start and execute them; they must be processed continuously in order to monitor the execution of the time.

For the runtime of user programs in the programmable controller, the following is valid for 1 K IL (1024 IL lines, approx. 4 kbyte):

ALU	100 % binary	65 % binary, 35 % words
ALU 150	approx. 4.0 msec	approx. 54.4 msec
ALU 821	approx. 4.0 msec	approx. 54.4 msec
ALU 286	approx. 1.7 msec	approx. 15.1 msec
ALU 011	approx. 0.9 msec	approx. 14.2 msec
ALU 061	approx. 0.5 msec	approx. 7.7 msec

However, the overall length of a user program is no firm indication of the scan time. Rather, only parts of the user program processed during the runtime contribute to the scan time, which can thus differ from scan to scan.

2.10 BSW Version 6.0, Module/New SFBs

2.10.1 Loadable Basic Software (BSW) Module Version 6.0

You can load the following modules with the combination of basic software version 6.0, AKF35 version 6.0 and ALU 021 or ALU 071:

Table 13 Modules of the loadable BSW V.6.0, AKF35 V.6.0

Module name	Use	Size [byte]
BUR	Blocks for module BUR 001	8 192
DBS	Blocks for module DBS 001	2 048
DOZ	Blocks for module DOZ 001	2 048
ERGBAU	Extension blocks	10 240
FERNBD	Remote operation, remote load	26 624
GPA*	Floating point arithmetic block	6 144
GPM*	Floating point measured value block	8 192
KOMQVL	Online comments, cross references	16 384
MASSFL	Mass flow blocks	2 048
MWVB	Measured value blocks	8 192
NOK	Blocks for module NOK 116	6 144
ONSTAT_1	Online status display for ALU 021	22 528
ONSTAT_3	Online status display for ALU 071	22 528
POSBAU	Blocks for modules POS 001/002/011	4 096
SA03	Blocks for modules SAI/SAA 103	8 192
TESYB	TESY blocks	43 008
TESYF	TESY functions	32 768
ZAE	Blocks for module ZAE 105	14 336
:		
:	can be extended with user modules	

* an arithmetic coprocessor is required on the ALU when these modules are used

2.10.2 SFBs without Configuration

No configuration of the basic software is necessary for the following blocks:

Table 14 SFBs outside the loadable BSW (on PADT)

Name	Number
for intelligent function modules	
VIP+	SFB1
VIPS+	SFB2
ISTD_POS	SFB61
EINR_POS	SFB62
HAND_POS	SFB63
AUTO_POS	SFB64
for closed-loop control	
KPID	SFB300
PID	SFB302
PIDP	SFB304
PI	SFB308
ZR	SFB310
DR	SFB315
PBM	SFB320
PDM	SFB325
PT1	SFB330
PT2	SFB331
DT1	SFB335
PDT1	SFB340
IB	SFB345
AB1	SFB350
AB2	SFB351
TZ	SFB355
STOE	SFB360
O–REG	SFB390
:	
: can be extended with user blocks	

☞ **Note** SFBs which are neither in Table 13 nor in Table 14 are part of the BSW base on EPROM (V. 6.0). They need not be configured separately.

Chapter 3
Operation

You can program with the Dolog AKF software on commerical IBM-compatible computers.
Data can be entered with a mouse and/or the keyboard.

3.1 Operation with the Mouse

In addition to the keyboard, line editor and cursor keys, you can enter data to the software with the mouse.
Please note that a parallel mouse (”Bus Mouse”) requires its own slot location in the PADT and it must be configured.
The manufacturer’s documentation describes the installation of the mouse.

Functions of the mouse in the software
The left button is used to call and to confirm a function, corresponding to the <Return> key.
The right button is used to abort, corresponding to the <Esc> key.

Expert The mouse can be driven at the COM1 interface if parameter /NOSPS is included in the call.
The mouse must be connected to the COM2 interface if parameter /NOSPS is not included in the call.

3.2 Operation with the Keyboard

3.2.1 US/German keyboard

The following keys are labeled differently on the German and English keyboards:

US / UK keyboard	**German keyboard**
<Esc>	<Eing Lösch>
<Ctrl>	<Strg>
<Home>	<Pos1>
<End>	<Ende>
<Prtsc>	<Druck>
<PgUp>	<Bild ↑>
<PgDn>	<Bild ↓>
<Ins>	<Einfg>
<Del>	<Lösch> or <Entf>

3.2.2 Keyboard Sections

The keyboard of a programming panel is divided into three different sections:

- **Typewriter keyboard and control keys**
  It is located in the middle (white keys) and is surrounded by grey control keys. Only the white keys cause a display of the printed symbol on the monitor if pressed.
  In the Dolog AKF software, they are used to enter data to the line editor, key macros and reference characters (see ”Special keys” chapter 3.2.8) and to enter text.

- **Function keys**
  They are of minor importance in the pulldown menu since all functions are selected with the mouse, reference characters or cursor keys.
  They are located above or next to the typewriter keyboard (depending on the model of the PADT) and are labeled <F1> ... <Fx>.

- **Numeric block**
  The cursor keys with the numerical block are located to the right of the typewriter keyboard. You can switch between the numerical and the cursor block here with the <NumLock> key.
  In the Dolog AKF software, mainly the cursor keys of the numerical block are required.
  They are used to select the different menu lines (see below).
  The numbers are used to call the graphic characters (e.g. in the title block):
  <Alt> + <number>.

3.2.3 Line Editor

The line editor supports you when entering text or numbers and comprises combinations of control keys and the typewriter keyboard keys.

It provides you with the following editor functions:

<Ctrl>+<A>, <Home>	Cursor to start of input line
<Ctrl>+<F>, <End>	Cursor to end of input line
<Ctrl>+<D>, < → >	Cursor to right
<Ctrl>+<S>, < ← >	Cursor to left
<Ctrl>+<M>, <Return>	Terminate input
<Backspace>, < ⇐ >	Delete character to left
<Del>	Delete character under cursor
<Ctrl>+<Y>	Delete input
<Ctrl>+<R>	Restore original text
<Ins>	Insert/overwrite switchover

If the first character of the input is a normal key (ASCII character), the input field is automatically deleted. The above-mentioned <Ctrl> keys maintain the contents of the input field.

The input is aborted and the previous text restored again with the <Esc> key.

You can insert stored text (e.g. symbols, comments, hardware addresses, file names and parts thereof) with the key macros (see below).

3.2.4 Screen Copy, ”Screensave”

The complete current contents of the screen can be copied to a file with this function. A printer need not be connected. The file is stored in the directory of the current station in IBM-ASCII format. The copy (file) can then be processed with a text editor.

Step Press the <Alt>+<F2> keys simultaneously to make a copy of the screen contents.

The first image is saved under the name ”BILD0.BLD”. Image 2 is called ”BILD1.BLD” etc.

Note After leaving Dolog AKF and restarting, the image counter again begins with ”BILD0.BLD”. Previous copies with the same name are overwritten.

3.2.5 Key input memory, ”Learn”

It is possible to combine up to 40 work steps (key inputs) with this function.

Step 1 Press the keys <Alt>+<F8> simultaneously to activate this function.

Step 2 Press the keys to be assigned input sequences simultaneously. You can assign <Alt>+<0>, <Alt>+<1> etc. up to <Alt>+<9>.

”Learn” appears in the status line.

Step 3 Select the required menues with the reference characters (up to 40 inputs per assigned key).

Step 4 Press <Alt>+<F8> simultaneously to end learning.

”Learn” disappears from the status line.

Step 5 Press the keys <Alt>+<0> or <Alt>+<1> etc. up to <Alt>+<9> simultaneously to execute the stored procedure.

Note The menues should be selected with reference characters to make the learned functions independent of the current menu bar position.

3.2.6 Key Macros

The editing of symbols, comments, hardware addresses, etc. is facilitated here. Absolute or symbolic addresses which are frequently used can be stored as text. Proceed as follows.

Step 1 Press the <Alt>+<F9> keys simultaneously to activate this function.

Step 2 Press the keys to be assigned text simultaneously. You can assign <Alt>+<A>, <Alt>+<B> etc. up to <Alt>+<Z>.

Step 3 Enter up to 19 characters of any text.

Step 4 Press the <Return> key to save.

Step 5 Press the <Alt>+<A>, <Alt>+<B> etc. up to <Alt>+<Z> keys simultaneously to call the stored text at the cursor location.

You can abort the input in step 4 with <Esc>.

Example: the text **Valve** was stored. It is inserted at the symbol/comment input (**Valve** 1 open, **Valve** 2 closed, **Valve** 1 semi, ...).

There is a separate text storage in the editor for symbols and comments (see also page 159).

3.2.7 Auto-Repeat Function

Most of the keys of a PADT are equipped with the so-called auto-repeat function: When a key is pressed, the labeled symbol is displayed on the screen or a corresponding function is executed. If the key is held down for longer than about 0.5 sec, the function of this key is repeated up to 10 times per second until the key is released.
Therefore you should make sure that these key functions are not used accidentally.

3.2.8 Special Keys

The following keys are of great importance in the menues:

- **Reference characters**
  Reference characters are used to directly select and execute the menu using the typewriter keyboard. You can call the required function by entering the marked capital letter.
  You can define the colors of the marking yourself with ”Setup”, ”Colors”.

- **<Ctrl>+Reference characters**
  The individual functions can be called with <Ctrl>+Reference characters within an editor without having to open the menu.

- **Cursor keys**
  You can select the individual lines of a menu with the cursor keys (motion keys, cursor keys, direction keys) <↑>, <↓>, <←>, <→>.
  The cursor keys are also used to select the catch words in the helptext.
  Pressing a key (during editing) corresponds to:

<↓>	one line down (in the same column)
<↑>	one line up (in the same column)
<→>	one column to the right (in the same line)
<←>	one column to the left (in the same line)

  Note that the keys only respond if the key <NumLock> is switched off. You can move the cursor with the mouse completely freely when editing the block.

- **<Return> key, <Ctrl>+<Return>**
  The <Return> key (also <Enter>, <Accept>, <Cr>) is used to call or accept (translate) certain functions. You can also ”toggle” with it (see below).
  Call
  - within the help function: to call the selected catchword helptext
  - in the pulldown menues: to call the selected menu line (function)
  - in editors (blocks, equipment list, etc.): the windows for editing functions are called with <Ctrl>+<Return> or with <Return>.

  End, Accept
  - during network editing and in entries with the line editor, the <Return> key is used to accept the parameter just entered, the hardware address, the comment, etc.

- **”Toggle”**
  Many menu functions permit direct selection of defined initial values. For example, you can switch directly between ”off” and ”on” or ”IL”, ”LD”, ”FBD” (”toggle”) without having to enter the text.
  You can toggle as follows:

  1 Type in the specified reference characters or
  2 Move the cursor to the line with the cursor keys and toggle with <Return>

The <Return> key corresponds to the left button of the mouse.

- **<Esc> key**
  Every function or input in the Dolog AKF software can be aborted with the <Esc> key.
  The <Esc> key corresponds to the right mouse button

Caution If you are at operating system level, you can only abort with <Ctrl>+<C>. This is true for the functions in the menu ”Special” and in the menu ”Print”.

- **<Ins> key**
  This key is toggled between ”insert” and ”overwrite” text in the line editor, comments and titles.
  A vertical spread occurs automatically in the block editor with ”insert”.

Table of important keys
A list of the most important special keys and key combinations will be omitted here. The same key combination can have different functions. If you have questions, look in the glossary under ”Key Table”. The keys which can be used are explained together with their meaning in the description of the particular menu.

Chapter 4
Programming

4.1 General

This chapter describes programming with Dolog AKF software.
For your reference, this chapter is followed by the menu structure of the Dolog AKF software. This means that the individual software segments are described in the following order:

Dolog AKF main menu	Chapter 4.2
Edit	Chapter 4.3
Load	Chapter 4.4
Online	Chapter 4.5
Print	Chapter 4.6
Special	Chapter 4.7
Setup	Chapter 4.8

This symbol indicates how you can select the described function. The path always starts with the main menu.
e.g.:
- ”Special”, ”Copy Disk complete”, ”Start ”

means:
Select ”Special” menu,
Confirm ”Copy Disk complete” line with Return,
Confirm ”Start Copying Disk” line with Return. The copy is then started.

☞ **Note** Some letters are written in upper case in the following description. They indicate the reference characters:
e.g. Se**T**up

First you are shown how to select the required menu line (the ”path” from the main menu) and then what functions you can execute at this point (programming, documentation, backup etc.).

The ”path” or ”paths” indicate the current point in the menu, beginning with the Dolog main menu.

4.2 Dolog AKF Main Menu

4.2.1 Status line

The lowest line of the screen is called the status line.

The 78 characters of the status line are divided as follows:

```
12345678<=1234567=>01234567|90123456789012345|789012|45678901234|6789012345678
```

- 1 - 8 Station name
- 9 - 10 Separator: <=
- 11 - 17 Type of networking: RS232/Modnet 1/Modnet 2/none
- 18 - 19 Separator: => (only for Modnet networking)
- 20 - 27 PADT station (only for Modnet networking)
- 28 - 28 Separator: |
- 29 - 45 free
- 46 - 46 Separator: |
- 47 - 52 Learn message
- 53 - 53 Separator: |
- 54 - 64 Dolog AKF status: FORCE on/FORCE off/no AKF
- 65 - 65 Separator: |
- 66 - 78 PLC scan status: PC* Stopped/PC* active/No connection

4.2.2 Pulldown Menues

The individual menu lines can be selected and called up with reference characters, cursor keys and <Return> or the mouse (see chapter ”Operation of PADT”).

4.2.3 Help Function

The helptext relating to the selected menu line in the pulldown menu can always be displayed with the <F10> key.
This text may contain so-called keywords. These are terms for which there is a further explanatory helptext. They are selected with the cursor keys and called up with <Return>. They help to maintain the clarity of the helptexts. Usually the texts are memory aids for Dolog AKF beginners. The keywords can be distinguished by their contrasting colors. Presssing <F2> takes you back to the previously selected helptext.
Pressing <F10> again will take you back to the overview of the helptexts. The keywords which can be selected are listed here (there are no separate lines in the menu for this!).

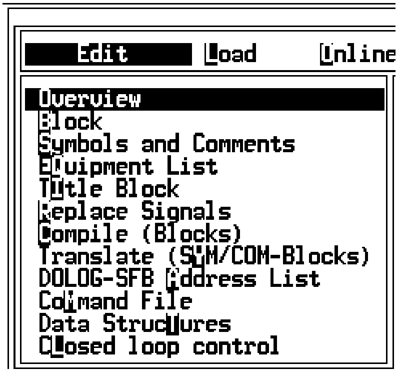
At the bottom edge of each help window you will find a line displaying the possible functions:

<↑>, <↓>, <←>, <→>	Select keyword
<Return>	Call helptext for keyword
<F1>	Call helptext for keyword
<Esc>	Abort and leave the help window
<F10>	Overview of helptexts
<F2>	Previous helptext is called
<PgUp>,<PgDn>	Scroll backwards, forwards within the helptext

4.3 Edit

The edit functions in the pulldown menu ”Edit” create your user program including the station documentation.

You can select and call the following menu lines:



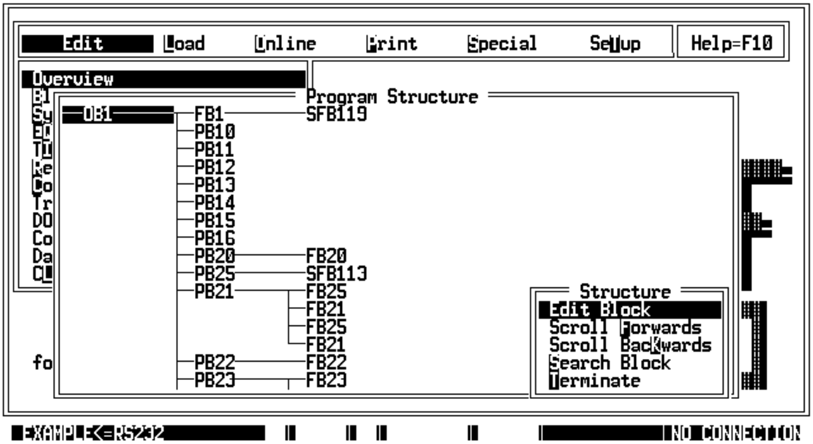
Caution **The station set up under ”Setup”, ”Station” is edited. Edit the equipment list before the blocks. If you want to program symbolically, first edit the symbols under ”Edit”, ”Symbols and Comments”.**

4.3.1 Overview

- ”Edit”, ”Overview”

This overview shows the complete program structure. The tree structure documents where and how often a particular block is called.

The survey looks like this after <Return> or <Ctrl>+<Return>:



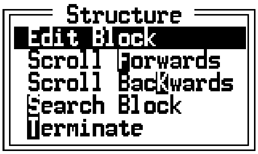
You can select the individual blocks with the cursor keys or the mouse.

Note If you assigned symbolic names for the blocks under ”Symbols and Comments” and the ”Addressing” is symbolic, you see the symbols when this function is called.

Program Structure

- "Edit", "Overview", <Return>

a) A menu with the following functions appears after pressing <Return> or <Ctrl>+<Return> :



The functions are called with <Return>, the reference characters or with the left mouse key.

b) You can call the functions with <Ctrl>+<Reference character> outside the menu, e.g. <Ctrl>+<**S**> for "**S**earch for Block".

Edit Block

- ”Edit”, ”Overview”, <Return>, ”Edit Block”

After selecting a block (e.g. with <PgUp>/<PgDn>), you can edit the (underlaid) block just selected with this function. Pressing <Return> takes you to the block editor.

Caution The standard function blocks (SFBs) are exceptions here. SFBs have a program defined by AEG. Therefore they cannot be selected directly and are only newly parametrized in the calling block.

After ”Terminate”, you return to the location of the editor call. The survey is then up-to-date.

Scroll Forwards/Backwards

- ”Edit”, ”Overview”, <Return>, ”Scroll Forwards”
  ”Edit”, ”Overview”, <Return>, ”Scroll Backwards”

You can page the screen up or down after selecting one of these functions.

This is possible outside the pulldown menu with the <Ctrl>+<F> and <Ctrl>+<K> keys.

Search for Block

- ”Edit”, ”Overview, <Return>, ”Search Block”

The block which you require is selected with this function. For a multiple call, the first one is shown.

Search for Block corresponds to the <Ctrl>+<S> key outside the pulldown menu.

Terminate

- ”Edit”, ”Overview”, <Return>, ”Terminate”

The program survey is terminated with the function and you are taken back to the edit menu.

4.3.2 Blocks

”Edit”, ”Block”

This pulldown menu is for the actual program creation. The individual blocks are programmed in the display mode you selected using the editors. First there is a description of the block editor, followed by an explanation of the individual menu lines.

After selecting the menu line ”Block”, a window appears in which

- you enter the block to be edited with the line editor and confirm with <Return> (e.g. OB1, PB55, FB368). Standard function blocks cannot be selected

Note If the selected block does not exist, it is created new (i.e. an empty screen appears after <Return>). In this case you can select an edit function after pressing <Return> again.

- a selection window containing all the blocks existing in the station is displayed after entering a space and <Return>. You can select the block to be edited with the cursor keys and <Return>.

Expert You can select ”Input Mode”, ”Addressing” and ”Address Mode” under ”Setup”, ”Station” or in the block editor under ”Presetting”.

Block Editors

The IL, LD and FBD editors are provided (depending on the input mode) as block editors.

The blocks OB, PB can be entered and represented in the special language Dolog AKF in IL, LD and FBD (Input Mode).
FBs can only be entered and represented in IL.

During editing of the user program (after Start Input), you can change the display mode as required with ”Presetting” (”toggle”). Networks are displayed on the screen in the selected display mode, regardless of the display mode in which they were created.

An exception is for example a network created in IL which is to be displayed in FBD but containing instructions (L, T, jumps) which cannot be displayed in FBD. Another exception is if the width of the screen is not sufficient for the number of elements to be displayed (an ”AND” with more than 7 inputs, an ”OR” with more than 16 inputs or several FBD elements cannot be displayed in LD).

In such cases the display is always in IL, regardless of the display mode defined.

During editing and representation of a network, the address mode (AEG/DIN) as well as the addressing (ABS/SYM) can be changed as required with ”Presetting”. The program is automatically checked for syntax errors after input.

Edit Organization Block OB1
A short description of the organization block can be found in Part V.

The following special features must be taken into consideration when editing an OB:

- The OB1 is processed cyclically. Each scan begins with the processing of network 001 and ends with the processing of the last network contained in the OB.
- PB calls, FB calls and possibly the user program reside in networks with consecutive numbering, starting with network 001.
- Each network may contain only one PB or FB call (except in IL).
- It calls PBs, FBs and SFBs conditionally or unconditionally. You can only enter a FB call if you previously edited the block.
- It may also contain program parts in IL, LD or FBD.
- Closed-loop control: The standard function block O-REG (SFB 390) must be called unconditionally in the OB1. This causes the organization block to be called by the system with the control functions (OBi) under interrupt control.

Edit Closed-loop Control Organization Block (OBi)

☞ **Note** If closed-loop control is configured, a second organization block is required.

- The closed-loop control functions are configured in this organization block (OB2 to OB999) in the program and function blocks. Standard function blocks for closed-loop control are also used.
- The OBi is processed under interrupt control. The basic sampling time is defined in O-REG (SFB 390) (time management table).

☞ **Note** Further information about the structure of closed-loop control programs can be found in the document:
”A350/A500
Regeln mit Dolog AKF
Benutzerhandbuch
A91V.12-2271963”[3)]

3) in german language

Edit Program Block PB

A short description of the program block can be found in Part V.

- A PB is a sequence of networks with consecutive numbering beginning with network 001.
- It can call PBs, FBs and SFBs conditionally or unconditionally. You may only enter a FB call if you previously edited the block.
- The same PB can be called more than once.
- It contains program parts in IL, LD or FBD.
- A block to be called is displayed in the OB (or another PB) as a rectangle (in LD/FBD) in the network. The PB number is above the rectangle. In conditional PB calls, the signal address of the call condition is to the left next to the rectangle. The call is also possible in the FB (in IL).
- A PB which is not called anywhere is never processed.

Conversion of a Program Block to Function Block:
Tested program blocks which the user has found to be suitable can be converted into function blocks with little effort.

Proceed as follows:

Step 1 Select the required program block with ”Edit”, ”Block”, ”Select Block”.

Step 2 Set the ”Input Mode” ”IL” with ”Presetting” for instruction list.

Step 3 Insert a further network (dummy network) before the first network and save the program block on the hard disk.

Step 4 Copy this program block within your station with ”Special”, ”Copy Files”, ”AKF Blocks” to a new function block (source block is the PB, target block is the new FB).
Possibly delete the old program block.

Step 5 Call the new function block with ”Edit”, ”Block” and select network 1.
This network is still empty.
Simply enter the new function block name (overwrite NW-end).
Possibly process the new function block by introducing formal parameters.

Step 6 Save the new function block.

Do not forget to delete the corresponding program block calls within your user program and to program new function block calls now.

Edit Function Blocks FBs

- FBs are used to create frequently recurring program parts as externally parametrized subroutines.

- FB1 ... FB999: user-defined function blocks
  In order to optimize time-critical processes, you should create your own solutions if possible. You can program up to 999 function blocks for this purpose.

- You must distinguish between the function block and the call of a function block. The FB contains a user program part. Calling a FB ensures that the FB is processed at runtime exactly when the call occurs in the user program. Before the block is processed, the formal operands of the FB are replaced by the given actual operands. A FB which is not called anywhere is never processed.

Structure

The program of the function block consists of a declaration and an instruction part. The declaration part is always in the first network of a FB. The instruction part follows in the subsequent networks.

Declaration Part

The name of the function block and the list of the formal operands including a specification of the type is included in the declaration part.

The following mask appears in network 1 after selecting an FB in the editor:

```
NAME:
     <IDENTIFIER TYPE ATTRIBUTE>
     :***
```

Creating the declaration part

Step 1 Enter the name of the function block in the first line.

Step 2 Open the first/next identifier line with <Return>.

Step 3 Insert the first formal operand in the first line denoted with "BEZ" (max. 4 letters/digits; the first character must always be a letter, the remaining positions may be defined as required).

Step 4 Call the selection window with spaces and <Return> in the column "Type".

Step 5 Select the required type using the cursor keys and confirm the selection with <Return>.

Step 6 Define in the column "Attribute" whether the formal operand is to be used as an input parameter (I) or output parameter (O) of the block. (You can delete existing entries with <backspace>.)

Step 7 Execute Step 2 to Step 6 for all other lines.

Step 8 Save the correct entries with <Ctrl>+<T>.

An example of a declaration part follows.

```
NAME  :  EXAMP1
         <IDENTIFIER  TYPE       ATTRIBUTE>
IDT   :  OP1          M          I
IDT   :  OP2          AABB       I
IDT   :  OUT1         MW         O
IDT   :  OUT2         M          O
IDT   :  ERR          Q          O
      :  ***
```

The meanings of the setpoint types are:
I : Input
Q : Output
M : Marker
MW: Marker Word
MD: Double word marker
MF: Floating point word marker
TN: Node number (number in equipment list)
Submenues also appear for this setpoint type
Data structures

An entry from the Attribute column (I/O) defines whether the parameter should appear in the graphic display as an input at the left edge of the block or as an output at the right edge of the block.

The option ”S” means that it is a system-defined setpoint type. This setpoint type is not displayed when the function block is called.

The option ”Q” means that the parameter resides in signal memory.

A suitable representative can be specified from each of these two groups.

The declaration part closes with ”***” (end of network).
In any subsequent modification, parameters may be deleted or inserted. The name of the formal operand can be changed at any time. If there are modifications, all the calls must be processed again.

Instruction Part

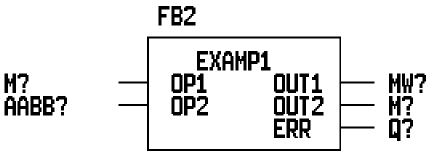
The instruction list, which defines the algorithmic relationship between the formal operands in the declaration part, is included in the instruction part. The name of the formal operand must always be preceded by an ”=” character in the instruction list. A number which specifies the nesting depth may be included to the right of the instruction list. The instruction list always ends with a ”:BE” (block end).

The following table shows parts of the instruction part of the above-mentioned function block ”EXAMP1”.

```
: O(
: A        =OP1              1
: A        =OP21.1           1
: A        =OUT2             1    Definition of nesting depth
: )                          1
: =        =ERR
: ***
```

Figure 10 Instruction part of a user-defined function block

Call



An FB can be called from another FB, PB or OB.
The same FB can be called more than once.
The FB is then displayed in LD/FBD as a rectangle in the network if called from a PB or OB . Its identifier is displayed again in the rectangle in abbreviated form. The input formal operands are at the left of the rectangle and the actual operands are to the left outside the rectangle. A possible condition is above the rectangle. At the right in the rectangle are the output formal operands and at the right outside the rectangle are the actual operands.

The call with the operations BC or BCC follows in IL.

If formal parameters are modified, deleted or inserted when editing the declaration part, all the function block calls must be programmed again. The global cross reference list can be used to determine where in the user program there are function block calls of the relevant function block.

It is also possible to call a function block from the instruction part of another function block (nesting, recursion). The called function block may contain its formal parameter as actualoperands in the instruction part of the calling function block. In this way actual operands can be passed on to the deepest level of nesting (10) while the user program is executing.
Tested program blocks which the user considers to be suitable can be converted into function blocks FB1...FB999 with a minimum of effort (see PB).

(Standard function blocks SFBs)

SFB1 ... SFB999:	Standard function blocks
SFB1 ... SFB499	are provided by Dolog AKF.
SFB500 ... SFB998	user-defined SFBs

The SFBs have different features. You can use them as often as required and need only parametrize them.
Basically, 999 SFBs are possible.

Data structures are also used to parametrize SFBs as of version 6.0.

☞ **Note** Data structures can currently only be used for: SFB1–2, SFB61–64, SFB300–390.

The method for using AEG-defined data structures is defined in the following steps.

- First use the tables in Part IV to define the data structures of your SFB.

- If you want to position with POS102 or POS112 or want to use closed-loop control, you can import suggestions for symbols, initial values and comments. Call the ”Special”, ”Import”, ”Symbols and Comments” function and enter POS.ASD or REGELN.ASD as the file to be imported.

- Assign symbols and possibly initial values per SFB call in a sample of the data structure (e.g. APOA**1**) to all the elements (e.g. APOA**1.1** to APOA**1.13**) under ”Symbols and Comments”. If you have imported the above files, suggestions will be made for the first sample. You can overwrite the presettings at any time.

- Enter a sample of the corresponding data structure (e.g. APOA**1**) for each SFB.

Display Modes IL, LD, FBD (Input Mode)

In the display ode (see also ”Input Mode”), you can select either the instruction list (IL), ladder diagram (LD) or function block diagram (FBD). The individual special languages in the Dolog AKF block editor will be discussed below:

Instruction List IL
The instruction list is a standardized display mode in alphanumeric form and corresponds to DIN 19239.

An instruction list consists of several instruction lines. It contains a part for labels (before the colon), a colon, the operation and the operand

: Operation Operand

OBs, PBs and FBs can be programmed in the IL.

Rules for Structure of Instruction List

Network Start / End

Example:

```
:   A..., O... or LD..
:
:   ***
```

Each network of the instruction part must begin with the operations A, O or LD and be terminated with "***" (end-of-network character).

Initializing Pulse

The initializing pulse is the startup marker "NORM" M1.1. The value of the startup marker is only 1 for the first scan.

Use: To standardize the actual values of timers and counters, when switching on the power supply or starting the program. For programming memory with preferred state after power-on, see "memory".

Initial Values

Initial values are entered under "Edit", "Symbols and Comments".

Markers

Network 001

```
:   A   I18.1
:   A   I18.2
:   O   I18.3
:   =   M5.12
:   ***
```

Markers (binary intermediate results) store results of logic operations that can be used at other points in the program. They must be defined before scanning. For nesting one works with markers or bracketed operations.

Network 002

```
:   A   I18.4
:   A   M5.12
:   =   Q17.1
:   ***
```

Bracketed Operations

Example:

```
: A  (       Bracketed operations can be used instead
: A  I18.1   of markers.
: A  I18.2   After an ”open bracket” instruction, the next
: O  I18.3   instruction must begin with A or O as at the
: )          program start.
: A  I18.4   The maximum nesting depth for bracketed operations is
: =  Q17.1   13. The number of ”open bracket” operations must be
: ***        equal to the number of ”close bracket” operations.
```

Logic Sequence

Example:

```
: A  I18.1   Logic operations follow a ”AND-before-OR”
: A  I18.2   rule as in Boolean algebra.
: O          i.e. superfluous brackets may not be entered.
: A  I18.3   The IL here follows the expression
: A  I18.4   (I18.1 ∧ I18.2) ∨ (I18.3 ∧ I18.4) = Q17.1.
: =  Q17.1
: ***
```

Standardizing Operations

Instructions with standardization

Example:

```
: A  I18.1   The result of the logic operation is standardized with
: S  M5.19   operations such as S..,R... The result cannot be used
: A  I18.2   immediately for the next instruction. The following
:            instruction must therefore always begin with the operation
:            A or O. The same method is valid for the instructions A(,
***          O(, AN(, ON(.
```

Instruction without Standardization

```
:  A  M5.4
:  A  M5.5   The result of the logic operation AM5.4, AM5.5 is not
:  =  M5.7   standardized for assignment ”=”, i.e. it is available for further
:  =  M5.8   assignments (up to 16 multiple assignments
:  =  M5.9   are possible).
:  A  I18.3  The register is first standardized before the subsequent
:            ”AND” operation is executed, i.e. deleted so that the previous
:  ***       result of the logic operation can no longer be used.
```

Mixed Logic Operations

Logic operations with bit- and word operands and those with different word operands (e.g. marker double words with marker floating point words) are not permitted.

Logic Operations with Constants

Constants can be used in IL. The values are displayed as constants (e.g. C30) during representation in LD or FBD, but cannot be edited in these display modes.

Memory Operations

It is essential to adhere to the order of the IL instructions. The preferred state after power on M1.1 or additional instructions before the ”=” assignment are optional.

If there are no additional assignments before the ”=”-assignment, marker M17.* can also be replaced by Q17.* in the examples below.

Example: SR-memory, dominant reset, preferred state after power on 1

```
: A   I18.1
: O   M1.1    The system marker M1.1 has a ”1” signal in the first
: S   M17.1   program scan, therefore preferred state after power on 1
: A   I18.2   The reset condition is processed after setting, thus
: R   M17.1   dominant reset.
: A   M17.1   Logic operation ”AND” necessary for output or marker
: =   Q17.1   before actual assignment, otherwise syntax error
: ***         occurs when ”terminating” the block.
```

Example: RS memory, dominant set, preferred state after power on 1

```
: A   I18.3
: R   M17.2
: A   I18.4
: O   M1.1    System marker M1.1 has a ”1” signal in the first
              program scan, therefore preferred state after power on 1.
: S   M17.2   Set condition follows reset condition, therefore domi-
              nant set.
: A   M17.2   Logic operation ”AND” necessary for output or marker
: =   Q17.2   before actual assignment, otherwise syntax error
: ***         occurs when ”terinating” the block.
```

Counters

You must adhere to the order of the instructions CTU/CTD, S, LD and R. Additional instructions may be included between these instructions.

Example: CTU Counter up

:	A	I18.1	Count pulse input (count with 0 → 1 edge at I18.1) Counter output set to ”1” with first edge.
:	CTU	MW200	MW200, Counter up
:	A	I18.2	Setpoint value loaded with MW201 with ”1”-signal on I18.2
:	S	MW200	
:	LD	MW201	
:	A	I18.3	Reset input (actual value and count output are set
:	R	MW200	to ”0” with ”1”-signal on I18.3).
:	=	Q17.1	For actual value = setpoint value: ”0”-signal on Q17.1.
:	***		

Example: CTD Counter down

:	A	I18.1	Count pulse input (count with 0 → 1 edge on I18.1 Counter output is set to ”1” with first edge.
:	CTD	MW202	MW202, Counter down
:	A	I18.2	Setpoint value loaded with marker word MW203
:	S	MW202	with ”1” signal on I18.2
:	LD	MW203	
:	A	I18.3	Reset input (actual value and counter output set
:	R	MW202	to ”0” ”with 1”-signal on I18.3).
:	=	Q17.2	For actual value = 0: ”0”-signal on Q17.2
:	***		

Timers

The function and runtime behavior of the timers corresponds exactly to the FBD elements of the FBD or LD editor during IL input. 5 time function blocks can be selected. The timer diagrams can be found under ”FBD Elements”. You must adhere to the specified order of the instructions. Additional commands may be included between the instructions.

Example: TP Pulse (Monoflop)

```
: A    I18.1    Start (input signal) of timing element MW500.
: TP   MW500
: LD   C30      Load constants C30 for timing setpoint
: A    I18.2    Reset timer block
: R    MW500
: =    Q17.1    With 0 → 1 edge on I18.1: ”1”-signal on Q17.1.
: ***
```

TEP Extended pulse

```
: A    I18.1
: TEP  MW502
: LD   MW503
: A    I18.2
: R    MW502
: =    Q17.2
: ***
```

TON Switch-on delay

```
: A    I18.1
: TON  MW504
: LD   MW505
: A    I18.2
: R    MW504
: =    Q17.3
: ***
```

TS Stored switch-on delay

```
: A    I18.1
: TS   MW506
: LD   MW507
: A    I18.2
: R    MW506
: =    Q17.4
: ***
```

TOF Switch-off delay

```
: A    I18.1
: TOF  MW508
: LD   MW509
: A    I18.2
: R    MW508
: =    Q17.5
: ***
```

Comparators

The comparators may only be used with operands of the same data type.
If representation in LD/FBD is dispensed with, arithmetic instructions are permitted between load and compare operations. There may be several instructions for bit operations before the ”=” assignment.

= equal

```
:   LD    MW300
:   EQ    MW301
:   =     Q17.1
:   ***
```

> greater than

```
:   LD    MW300
:   GT    MW301
:   =     M5.22
:   ***
```

< less than

```
:   LD    C20
:   LT    MW51
:   =     Q17.3
:   ***
```

<> not equal

```
:   LD    MW302
:   NE    MW303
:   =     Q17.4
:   ***
```

≥ greater than/equal

```
:   LD    MD330*
:   GE    MD332*
:   =     M5.23
:   ***
```

≤ less than/equal

```
:   LD    MF350*
:   LE    MF352*
:   =     Q17.6
:   ***
```

*   Reserve at least two words per operand for MD and MF (e.g. MD550/ MD552

Arithmetic (ADD, SUB, MUL, DIV)

All arithmetic operations must start with a load operation.

```
: LD    V1        There may be several arithmetic instructions one
: ADD   MW1501    after the other.
: SUB   MW1502
: T     MW1503
: ***

: LD    MW1511    In arithemtic instructions there is no
                  ”point-before-dash” rule.
: MUL   MW1512    The expression
: ADD   MW1510    <MW1510> + <MW1511> x <MW1512> =
: T     MW1520    <MW1520> must be programmed as here.
: ***
```

- The arithmetic may only be used with operands of the same data type.
- Brackets may not be used in arithmetic instructions. If intermediate results are to be used at a later time, intermediate markers must be introduced.
- To detect a division by zero, a comparator (divisor = 0?) must be included in the instruction list.
- To prevent arithmetic overflow, the operands must be checked for size. Example: Is the result of an addition less than one of the summands?
- Constants must be loaded with ”LD”.

Transfer

Bit → Word

The LB operations load (read into a register) a defined bit string (word). The register contents (binary) are transferred to a digital value register with the assignment operation, e.g. T MW1510.

The contents of the digital value register are always displayed decimally.

Note The operand following the load operation is the least significant binary signal of the bit string.

There is a syntax check of the validity of the bit string during input depending on the instruction list.

Network 001

```
  :  LB     M4.4      Transfer the values from bit string M4.4 ... M4.19 to
  :  T      MW1501    marker word 1501.
  :  ***
```

Word → Bit

The TB operations assign a bit string to the register contents (binary).

The transfer begins with the least significant bit of the register contents and ends with the operand following the transfer operation.

The start of the bit string is defined by the operand.

Network 002

```
  :  LD     MW1510    Transfer the contents of MW1510 to the bit string
  :  TB     M5.1      M5.1 ... M5.16
  :  ***
```

Formal Operands in FBs

A maximum of 45 input or output parameters may be defined as formal operands in the declaration part of a user function block, but in all there may not be more than 64 parameters. A data structure counts as a FB/SFB parameter.

Only formal operands which were previously defined in the declaration part are accepted in the IL of such FBs. A ”=” character must precede each formal operand during input.

A maximum of 24 jump labels per network is permitted within an FB.

Jumps

Entry of jump labels

You can use conditional and unconditional jumps within a network of an FB in IL. The label which is entered in the jump call must also be specified for the jump destination. Jump labels are defined as formal parameters with an ”=” character. An unconditional jump JI may be anywhere in the network, but the conditional jump JT is only possible in place of an ”=” assignment.
JT is executed for a ”1” signal of the condition.

The jump destination should be a load instruction for arithmetic or the start of a logic operation as the register is only standardized here. If the operation directly following a conditional jump is an output instruction, it is executed with the result of the logic operation of the jump condition. All others cause the register to be standardized.

Example: JI unconditional jump

```
      :  A  I18.1    Jump labels are entered to the left of the colon.
      :  A  I18.2    Move the cursor there (cursor keys or
      :  =  Q17.1    mouse). Jump labels have up to four
      :  J  =M001    characters. The first character
      :  ...         must be a capital letter. The other characters
 M001 :  A  Q17.1    may be digits, special chars. or letters,
      :  ...         but no spaces.
```

☞ **Note** Jumps in LD and FBD may not be represented.

NOP Instruction

The ”NOP” instruction is a placeholder for an output operation.

☞ **Note** Instruction lists with ”NOP” instructions cannot be represented in LD or FBD.

Ladder Diagram LD

The ladder diagram is a standardized graphic display mode (e.g. for circuit diagrams) and corresponds to DIN 19239.

OBs and PBs can be programmed in LD.
A maximum of 16 signals can be linked in parallel and seven serially as inputs in a ladder diagram for a single network. There is only one output per LD network and it cannot be negated. The value of the output signal can be applied to a mximum of 16 contacts.
In LD there are no jumps, only block calls.
There is a return to the calling point when the called block has been processed.
A program or function block call may only stand alone in an LD network.
When creating networks in LD, make sure that the elements may only be entered at certain cursor positions (see LD editor).

The following symbols can be selected during network creation in LD:

—] [—	Normally open contact
—]/[—	Normally closed contact
———|	Connection of parallel ladder digram lines
———	Continuation in parallel path without contacts
—( )—	Output
FBD elements:	Other elements which can be called in LD and FBD (memory, timers, counters and comparators) can be selected and called.

⚠ **Caution about FBD elements: Normally only one FBD element per network is possible in LD programming. Exceptions in the uppermost cross-path are several timers, counters or flip-flops next to one another.**

The Dolog AKF software makes a syntax check during network creation in LD, rejecting illegal entries.

Further information about network creation in a ladder diagram can be found in the LD editor and in ”Correction Mode” of the LD editor.

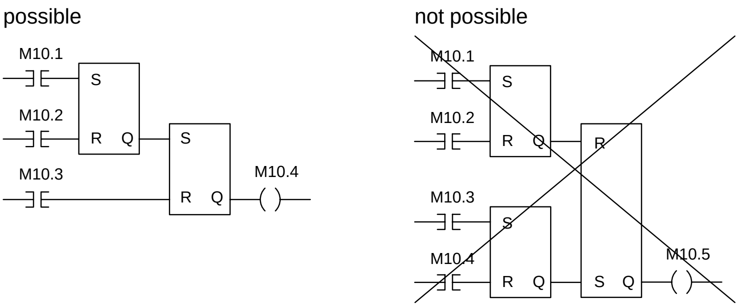


Figure 11 Input of FBD Elements in the Ladder Diagram

Function Block Diagram FBD

The function block diagram is a standardized graphic display mode and corresponds to DIN 19239.

OBs and PBs can be programmed in FBD.
A maximum of six FBD elements horizontally and 46 input signals plus a maximum of 12 FBD elements each with two inputs vertically are possible in one FBD network. There is only one output per FBD network. It cannot be negated.
The value of the output signal can be applied to a maximum of 16 contacts.
There are no jumps in FBD, only block calls.
There is a return to the calling point when the called block has been processed.
A program or function block call may only stand alone in an FBD network.
When creating networks in FBD, make sure that the elements only be entered at certain cursor positions.

The following symbols can be used in FBD for creating a network:

&	AND block
>=1	OR block
——┤	Input
——○┤	Input negated
FBD elements:	The elements which can be called in LD and FBD (memory, timers, counters and comparators) can be selected and called.

FBD Elements

The FBD elements are standard blocks which can be selected in the FBD or LD editor.

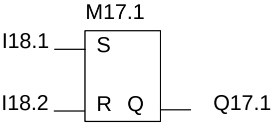
All AKF functions (FBD elements) have retentive characteristics, i.e. the signals and words remain unchanged in the programmable controller after power failure and return. To achieve initial state behavior (standardization after power return), the system marker M1.1 must be applied to the standardization input of the function.

The addresses MW100 ... MW799 must be used for the setpoint and actual values of the AKF timer and counter functions (value range 0 ... +4095).

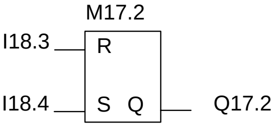
The FBD elements are explained below with an example.

Flipflops

SR flipflop
dóminant reset

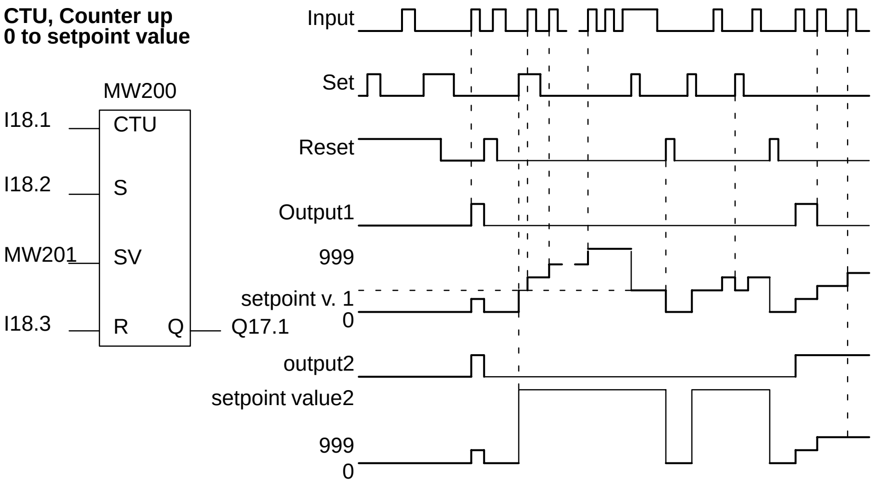


RS flipflop
dominant set



Counters

CTU, Counter up
0 to setpoint value



The counter continues to 999 after reaching the setpoint value, unless it is reset beforehand.

Output 1/Setpoint value 1:	Setpoint value is less than 999; the current value curse is shown
Output 2/Setpoint value 2:	Setpoint value is greater than 999; the current value curse is shown.

Figure 12 FBD elements and scan time diagram of the up counter

CTD, Counter down
Setpoint to 0

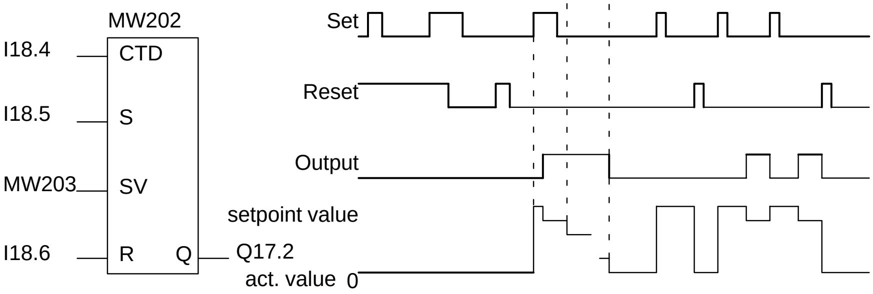


Figure 13 FBD elements and scan time diagram of the down counter

Timers

TP, timer pulse

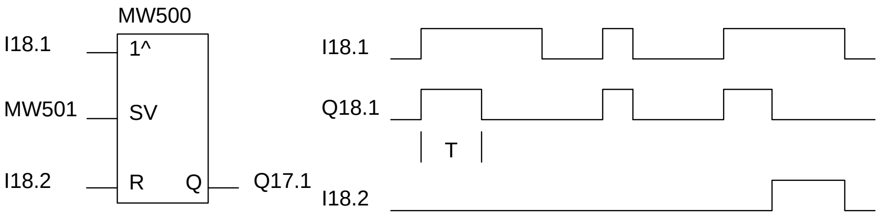


Figure 14 FBD element and scan time diagram of the pulse

TEP, extended pulse

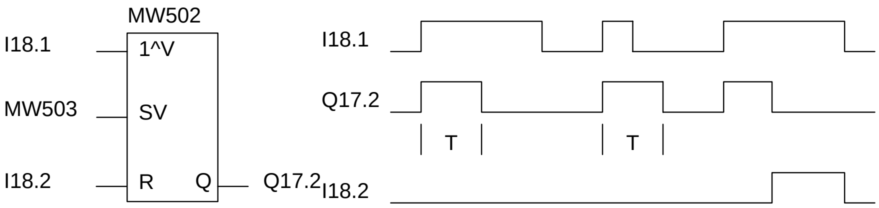


Figure 15 FBD element and scan time diagram of the extended pulse

TON, switch-on delay

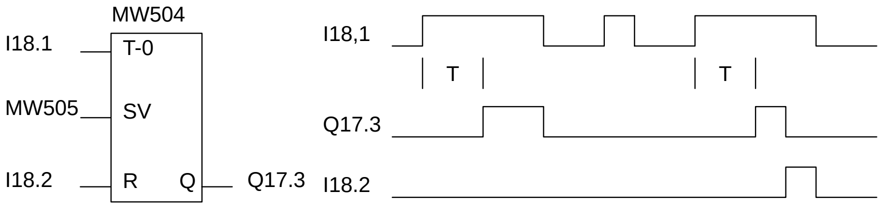


Figure 16 FBD element and scan time diagram of the switch-on delay

TS, stored switch-on delay

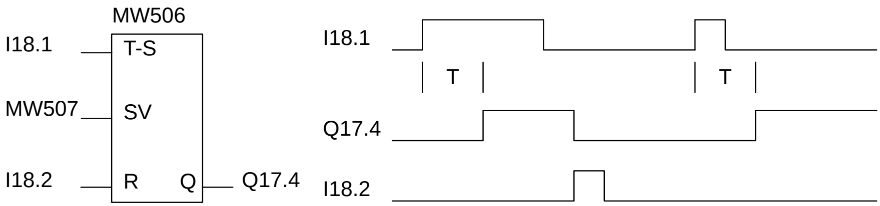


Figure 17 FBD element and scan time diagram of the stored switch-on delay

TOF, switch-off delay

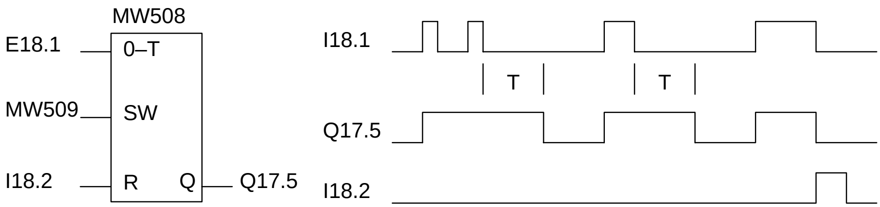
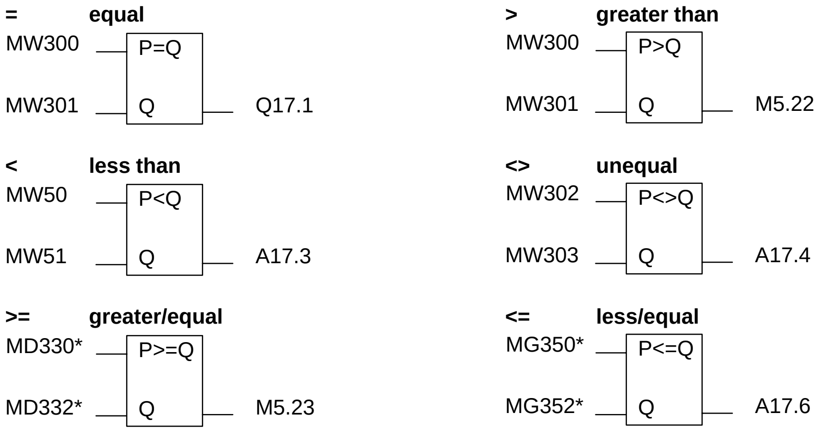


Figure 18 FBD element and scan time diagram of the switch-off delay

The timer functions have a constant time base of 100 msec. Internally they process 4000 count increments, so maximum times of 400 sec (4000 x 100 msec) can be obtained. Longer times can be obtained by connecting counter functions serially or with an extra timer function.
The timer functions are called software timers, i.e. it is not enough to process them just once to activate and execute them. They must be executed in each program scan.

Comparator



* Reserve at least 2 words per operand for MD and MF (e.g. MD550/ MD552

The contents of two words, two double words or two floating point words (and poss. constants) are compared. Six different comparators can be called. ”Word comparison”, ”Double word comparison” and ”Floating point word comparison” are distinguished here.

If the condition described in the FBD element is satisfied, output ”Q” = 1.

The Dolog AKF software makes a syntax check when creating the networks in FBD, rejecting invalid input.

Further information about network creation in the function block diagram can be found in the FBD editor and in ”Correction Mode” of the FBD editor.

Edit Block

- ”Edit”, ”Block”, Block, <Return>

If the block entered does not exist, it is newly created.

```
Select Block
OB1
```

To call the edit pulldown menu, press the <Return> key or <Ctrl>+<Return> (”Edit Network” menu) again.

```
Edit Network
Insert
Erase
Copy
Modify
Scroll Forwards
Scroll Backwards
Terminate (save)
Break
Signal Search
Network Search
Exchange Online
Dyn. Status Display
Presetting
```

You can also call the corresponding helptext in each step with <F10>.

You can abort the edit at any time with the <Esc> key.
All data entered up to this time is lost.

Editing in the Different Display Modes

You can program in three display modes. You can change the special language at any time. Some exceptions, however, must be observed.

The different editors are explained on the following pages.

There are menu lines or functions which occur at several locations in the editor or which are the same in several editors. The ”Edit Network” menu is an example of this. These positions are only described the first time they occur. The index of keywords can be used for further orientation.

Instruction list IL

IL Editor

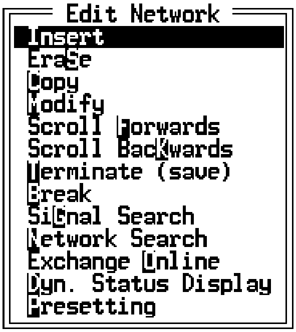
- ”Edit”, ”Block”, Block, <Return>
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Note You can still process an existing block in the required display mode at a later time. Pressing <Return> takes you to the menu ”Edit Network”. The structure of this menu is the same for all display modes. The common functions are described here for IL.

You are in the IL editor, which will be described below.

You can edit in IL
a) in the pulldown menu ”Edit Network” using the reference characters by selecting the menu line with cursor keys and calling with <Return> or using the mouse
b) with the <Ctrl>+<reference character> from the next higher editor level

a) The following functions are provided in the pulldown menu ”Edit Network”



b) if you did not select the menu ”Edit Network”, you can execute the functions with <Ctrl>+<reference character>, e.g. <Ctrl>+<**M**> for ”**M**odify”. The following keys also are in effect:

<PgDn>	scroll to next network
<PgUp>	scroll to previous network
<Esc>	abort input without saving

During network creation in IL, editing is line-by-line and in tabular form, i.e. certain tabulator positions are reserved for the IL instructions (operations), the operands and the 31 characters IL line comments (for FBs). These can be moved by presing the <tab> key or with the cursor keys.

In function blocks (FBs), each IL line can be assigned a maximum of 31 characters of its own comment. The line editor/typewriter keyboard with all its keys (also e.g. <backspace>) can be used within the comment line.

Edit IL, LD, FBD / Insert Network

- ”Edit”, ”Block”, Block, <Return>, ”Insert”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

You can insert a new network in the pulldown menu ”Edit Network” with this function.

Insertion is before the displayed network for IL, LD and FBD.

Exception: Insertion is not permitted before the first network of a FB.

You can call further editing menues in a new network with <Return> or <Ctrl>+<Return> (see also ”Modify Network” in the individual input modes).

Edit IL, LD, FBD / Erase Network

- ”Edit”, ”Block”, Block, <Return>, ”Erase”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

You can delete the displayed network in the pulldown menu ”Edit Network” with the function ”Erase”.

The last remaining network of a block cannot be deleted with this function. The whole block is then deleted under ”Special”, ”Erase Files”, ”AKF Blocks”.

Edit IL, LD, FBD / Copy Network

- ”Edit”, ”Block”, Block, <Return>, ”Copy”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

You copy a network of any source block of the station to the current network with this function. The following menu is opened with <Return>:

```
Network copy
Start Copying
Target-Block:        PB1
Source-Network:      1
```

- ❐ The ”Target block” is the source block from which the new network is to be copied. You can display a selection window with a space and <Return>.
- ❐ The ”source network” is the network to be copied. You can display all the networks of the source block with space and <Return>.
- ❐ The selected network is copied with ”Start Copy”. The new network is given the number of the network in which the copy is carried out (”current network”).

The previous network with this number and all subsequent networks are shifted back by one position.

Edit IL, LD, FBD / Modify Network

- ”Edit”, ”Block”, Block, <Return>, ”Modify”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

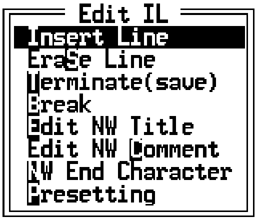
The correction mode of the block editor is set using this function in the pulldown menu ”Edit Network”.
You can then modify the network displayed on the screen.

IL Editor / Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

You leave the menu ”Edit Network”, ”Modify” (existing block) or the menu ”Edit IL” (new block) in correction mode of the IL editor.

a) The pulldown menu ”Edit IL” has the following functions:



b) If you did not select the ”Edit Network” menu, you can execute the functions with <Ctrl>+<reference characters>, e.g. <Ctrl>+<**P**> for ”**P**resetting”. The following keys also have effect:

<↑>, <↓>, <←>, <→>	Move cursor
<Ins>	Insert line before the current line (Cursor to colon of current line)
<Del>	Delete line (Cursor to colon of line)
<tab>	Next input field (tabulator position)
<backtab>	Previous input field (tabulator position)
<Esc>	Abort input without saving

If you press the <Return> key, you can also execute the listed functions in the pulldown menu ”Edit IL” with the menu lines.

Edit IL / Insert Line in Correction Mode

- "Edit", "Block", Block, <Return>, "Insert", <Return>, "Insert Line"
- "Edit", "Block", Block, <Return>, Existing Block, <Return>, "Modify", <Return>, "Insert Line"
- "Edit", "Overview", Select Block, <Return>, "Edit Block",..

You can insert instruction lines at any location in IL networks with this function by selecting the corresponding menu line and <Return>, with the reference characters or with the mouse.
You can insert lines with the <Ins> key outside the pulldown menu.

Insertion is always before the line to which the cursor is pointing.

Example:

: A M 1.1	: A M 1.1	: A M 1.1
: O I 2.12	:	: A M 5.1
: = Q 3.2	: O I 2.12	: O I 2.12
: ***	: = Q 3.2	: = Q 3.2
	: ***	: ***
insert before 2nd line	line is inserted	enter instruction

Edit IL / Erase Line in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Erase Line”
- ”Edit”, ”Block”, Block, <Return>, Existing Block, <Return>, ”Modify”, <Return>, ”Erase Line”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

You can delete the current instruction line in IL networks with this function.
The line to which the cursor is pointing is deleted.
If you use the special key, the cursor must point to the colon of the line in order to delete the current line (otherwise only a letter is deleted).

Edit IL, LD, FBD / Terminate in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Terminate”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Terminate”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Correction mode of the editor and thus network correction is terminated with this function. The inserted or modified network is accepted.

Caution The modifications will be lost if you now abort in the menu ”Edit Network” with ”Abort” or <Esc>. ”Terminate” must also be selected here to translate and store on the hard disk.

Edit IL, LD, FBD / Break in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Break”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Break”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Correction mode of the network is aborted with this function or with <Esc>. The modifications are rejected and the old state is maintained.

The abort corresponds to the <Esc> key outside the pulldown menu.

Edit IL, LD, FBD / Network Title in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Edit NW-Title”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Edit NW-Title”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

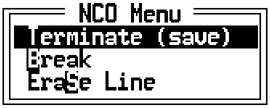
A network title containing up to 32 characters may be entered in the last editor line.

Edit IL, LD, FBD / Network Comment in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Edit NW-comments”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Edit NW-comments”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

A maximum of 15 x 60 characters of comments can be entered to the network in one window. The lines may have any structure. Each line is terminated with <Return>.

The following window is called with the key <Ctrl>+<Return>:



If you did not select the ”Edit Network” menu, you can execute the functions with <Ctrl>+<reference characters>, e.g. <Ctrl>+<**T**> for ”**T**erminate”. The following keys also have effect:

<↑>, <↓>, <←>, <→>	Move cursor
<Ctrl>+<S>	Cursor left
<Ctrl>+<D>	Cursor right
<Ins>	Insert mode on/off (insert character)
<Del> , <Ctrl>+<G>	Delete character under cursor
<backspace>	Delete character before cursor
<Home>	Cursor to start of line
<End>	Cursor to end of line
<Esc>	Abort input without saving

Edit IL / NW End Charcter in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”NW End Character”
- ”Edit”, ”Block”, Blocks, <Return>, ”Modify”, <Return>, ”NW End Character
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

A network end character is entered at the cursor position in the current network with this function.

Caution Subsequent lines of the network are deleted.

Edit IL, LD, FBD / Presetting in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Presetting”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Presetting”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

You can modify the basic settings of the editor here.

The presetting menu provides the following functions in correction mode:

Select Editor Setting		
Cursor Positioning	Horizontal	(horizontal/vertical/off)
Output Monitoring	On	(on/off)

Edit IL,LD,FBD / Presetting / Cursor Positioning

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Presetting”, ”Cursor Positioning”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Presetting”, ”Cursor Positioning”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

With automatic cursor positioning, you can set the direction in which the cursor should move between the question mark fields during input to LD/FBD by toggling.

Horizontal position:
In ladder diagram address input, the cursor moves horizontally from one element to the next in a network line.

Vertical position:
In ladder diagram address input, the cursor moves vertically from one element to the next in a network column.

Off position:
The cursor is moved character-by-character on the screen.

Output Monitoring

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Presetting”, ”Output Monitoring”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Presetting”, ”Output Monitoring”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

You can toggle between ”on” and ”off”.
If output monitoring is ”on”, the address assignment is monitored during editing. A message is output if an output address which was already used is entered again.
The programmer is thus informed of multiple output assignments. The messages must be acknowledged.

Edit IL, LD, FBD / Network Scroll Forwards

- ”Edit”, ”Block”, Block, <Return>, ”Scroll Forwards”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..
- ”Online”, ”Dyn. Status Display”, ”Online Recording” or ”Single Shot Processing” or ”Triggered Recording”, <Return>, ”Start Display”, <Return>, ”Scroll Forwards”

The next network is called when this function is selected. If networks cannot be displayed in the defined display mode (LD/FBD), there is automatic switch to IL.

Scroll forwards corresponds to <Ctrl>+<F> or the <PgDn> key outside the pull-down menu.

Edit IL, LD, FBD / Network Scroll Backwards

- ”Edit”, ”Block”, Block, <Return>, ”Scroll Backwards”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..
- ”Online”, ”Dyn. Status Display”, ”Online Recording” or ”Single Shot Processing” or ”Triggered Recording”, <Return>, ”Start Display”, <Return>, ”Scroll Backwards”

The previous network is called when this function is selected. If networks cannot be displayed in the defined display mode (LD/FBD), there is an automatic switch to IL.

Scroll backwards corresponds to <Ctrl>+<K> or the <PgUp> key outside the pulldown menu.

Edit IL, LD, FBD / Terminate (save)

- ”Edit”, ”Block”, Block, <Return>, ”Terminate (save)”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

The edit is terminated when this function is selected. There is a syntax check for the entered block and it is compiled and saved on the hard disk.

Edit IL, LD, FBD / Break

- ”Edit”, ”Block”, Block, <Return>, ”Break”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

The edit is aborted after inquiry when this function is selected. All modifications are lost after acknowledgement of the inquiry.

Abort corresponds to <Ctrl>+<B> or the <Esc> key outside the pulldown menu.

Edit IL, LD, FBD / Network Signal Search

- ”Edit”, ”Block”, Block, <Return>, ”Signal Search”
- ”Edit”, ”Symbols and Comments”, Block, <Return>, <F9>, ”Search Function”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..
- ”Online”, ”Dyn. Status Display”, ”Online Recording” or ”Single Shot Processing” or ”Triggered Recording”, <Return>, ”Start Display”, <Return>, ”Signal Search”

You can search for signals within the block using the search function. (There is no search in the current network.)
Absolute or symbolic signal names may be entered. If the software finds the signal, it is displayed at the lower edge of the screen.

The search direction (forwards or backwards) can be set as an option with the characters ”+” or ”-”. If no entry is made, there is a forwards search.

You can activate this function in the window by pressing the <Return> key. You close the window again with the <Esc> key.

The following entries are possible for settings of:

	DIN	AEG
Inputs	I2.1 .. I160.32	I2A2 .. I160E32
Outputs	Q2.1 .. Q160.32	Q2A2 .. Q160E32
Markers	M1.1 .. M313.16	M1 .. M10 000
Marker word	MW1 .. 10 000	
Marker double word	MD1 .. 9 999	
Marker floating point word	MF1 .. 9 999	

PBs, FBs, data structures and formal operands (e.B. =OP1) can also be searched for.

Recommendation: see page 47.

Symbol names can be entered with the appropriate address mode.
The search direction is defined by a prefixed <+> (forwards) or <-> (backwards).

Edit IL, LD, FBD / Network Search

- ”Edit”, ”Block”, Block, <Return>, ”Network Search”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..
- ”Online”, ”Dyn. Status Display”, ”Online Recording” or ”Single Shot Processing” or ”Triggered Recording”, <Return>, ”Start Display”, <Return>, ”Network Search”

You can search for networks within the block with this search function.

Enter the network number if you know it.
Enter a space or ”0” and <Return> if you want to display a selectionwindow of all the networks in the block (with NW title if it exists). You can then select a certain network with the cursor keys and <Return>.

You can activate the function in the window by pressing the <Return> key.

You can close the window again with the <Esc> key.

Edit IL, LD, FBD / Exchange Online

- ”Edit”, ”Block”, Block, <Return>, ”Exchange Online”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

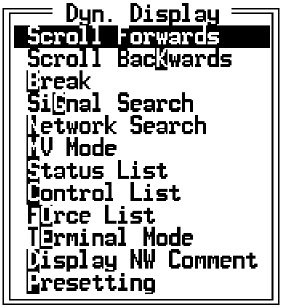
Caution This is only possible for programs running in the programmable controller. Therefore this function will not be described here. However, it can also be executed here if individual blocks are to be re-edited. Please read the chapter ”Load”, ”Exchange Online”.

Edit IL, LD, FBD / Network Dynamic Status Display

- ”Edit”, ”Block”, Block, <Return>, ”Dynamic Status Display”
- ”Edit”, ”Overview”, select Block, <Return>, ”Edit Block”,..

Caution This is only possible for programs running in the programmable controller and will therefore not be described here. The functions are described under ”Online”. However, the status display can also be executed here if individual blocks are to be re-edited.

The following functions can be used in the dynamic status display.



The signal states are masked into the current network dynamically after <Return> for ”PC* in scan”.
You can return to editing mode with <Esc>.
The description can be found in the chapter ”Online”, ”Dynamic Status Display”.

Edit IL, LD, FBD / Network Presetting

- ”Edit”, ”Block”, Block, <Return>, ”Presetting”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

You can change the basic settings of the editor here.

The following functions can be selected:

Select Editor Setting		
Input Mode	FBD	(LD/FBD/IL))
Addressing	ABS	(absolute/symbolic)
Cursor Positioning	Horizontal	(horizontal/vertical/off)
Output Monitoring	On	(on/off)
Address Mode	DIN	(DIN/AEG)

Cursor positioning and output monitoring are provided in editor correction mode in the presetting menu.

The presetting corresponds to the <Ctrl>+<P> keys outside the pulldown menu.

Input Mode

- ”Edit”, ”Block”, Block, <Return>, ”Presetting”, ”Input Mode”
- ”Setup”, ”Station”, ”Input Mode”

You define the display mode in the editor here. You can toggle between IL, FBD and LD.

Note Networks which cannot be displayed in LD or FBD are automatically displayed in IL.

Addressing

- ”Edit”, ”Block”, Block, <Return>, ”Presetting” ”Addressing”,
- ”Setup”, ”Station”, ”Addressing”

You can address all inputs, outputs, markers, marker words... absolutely (e.g. I 2.15, M 310.17) or symbolically if you previously created the SYM/COM block. You can toggle between them.

Address Mode

- ”Edit”, ”Block”, Block, <Return>, ”Presetting” ”Address Mode”,
- ”Setup”, ”Station”, ”Address Mode”

You can toggle between two address modes for markers, inputs and outputs:
1 according to DIN 19239 or
2 according to the hardware terminal designation (AEG).
The assignments are described in the chapter Validity Scopes.

Rules for the use of functions processing bit strings (e.g. LB, TB, SFB119, SFB120).
The following is valid for front connection modules:
When specifying bit strings (e.g: parameter BK, BL of SFB119), you should use the tables (chapter 2.2 following).

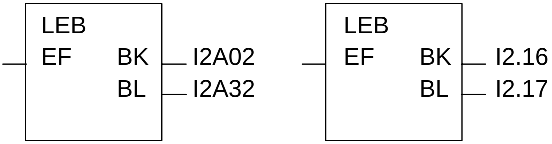
Example: Pin string I2.1 to I2.32 of front connection module should be deleted with SFB119.

Step 1 Pin”x.1” in the column DIN leads in the same line of the table to pin”x.A02” at DEP/DAP0xx.

Step 2 Pin”x.A02” at DEP/DAP11x leads in the same line of the table to ”x.16” in the column DIN and thus obtains ”I2.16”.

Step 3 The same method is used for pin ”I2.32” and one obtains after Step 1 and Step 2 pin ”x.17” and ultimately ”I2.17”.

Example: Calls of equal value for LEB (SFB119), front connection technique



Caution Because of the different addressing for back and front connection techniques, you cannot change between back and front connection technique in the equipment list at a later time during bit string processing.

LD Editor

- ”Edit”, ”Block”, Block, <Return>
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Note If you select an existing block in the required display mode, you can still process it at a later time. You enter the menu ”Edit Network” after pressing <Return>. Its structure is the same for all modes of display. The common functions are described in IL.

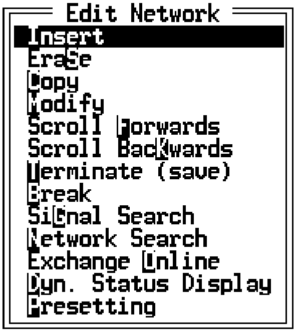
You are in the LD editor, which will be described below.

You can edit in LD

a) in the pulldown menu ”Edit Network” using the reference characters by selecting the menu line with the cursor keys and calling with <Return> or using the mouse,

b) with <Ctrl>+<reference character> from the higher editor level

a) The following functions are provided in the pulldown menu ”Edit Network”:



b) if you did not select the menu ”Edit Network”, you can execute the functions with <Ctrl>+<reference character>, e.g. <Ctrl>+<**M**> for ”**M**odify”. The following keys are also used:

<PgDn>	Scroll to the next network
<PgUp>	Scroll to the previous network
<Esc>	Abort input without saving
< ↑ >	Shift screen up
< ↓ >	Shift screen down

If you press the <Return> key, you can also select the above-mentioned functions with the menu line of the pulldown menu ”Edit Network” appearing at the lower right of the screen.

When creating networks in LD, note that elements may only be input at certain cursor positions.

These positions must been selected with the cursor keys. The cursor must point to the circuit diagram when editing.

Insert LD element:	to the right of the cursor
Insert FBD element:	cursor must point to the first character of the contact (poss. spread first)
Delete LD element:	to the right of the cursor
Delete FBD element:	elements of the same type can be overwritten if the cursor points to the first input (ladder diagram line)

Automatic cursor positioning with the menu ”Presetting” is possible for the address input.

The address input starts with the first question mark.

LD Editor / Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

You enter correction mode of the LD editor from the menu ”Edit Network” (existing block) or from the menu ”Edit LD” (new block, after entering the first element).

a) Three pulldown menues ”Edit LD” can occur during editing:

```
Edit LD
N/O: ---] [---
N/C: ---]/[---
PB/FB/SFB Conditional
PB/FB/SFB Unconditio.
Break
FBD Element
Presetting
```

```
Edit LD
Spread Horizontal
Spread Vertical
N/O: ---] [---
N/C: ---]/[---
R: ---( )---
FBD Element
Erase
Terminate (save)
Break
Edit NW Title
Edit NW Comment
Presetting
```

```
Edit LD
N/O: ---] [---
N/C: ---]/[---
V: ---|
H: ---------
```

b) You can carry out the functions with <Ctrl>+<reference character>, e.g. <Ctrl>+<**F**> for ”**F**BD Element”. The following keys are also used:

<↑>, <↓>, <←>, <→>	Move cursor
<Del>	Delete element
<PgUp>	Shift image up line by line
<PgDn>	Shift image down line by line
<Ctrl>+<PgDn>	Shift image up page by page (scroll forwards)
<Ctrl>+<PgUp>	Shift image down page by page scroll backwards)
<Letter>	Change contact address
<Esc>	Abort without saving

You can also carry out the above-mentioned functions with the menu lines in the different pulldown menues ”Edit LD” by pressing the <Return> key.

Edit LD / Horizontal Spread in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Spread Horizontal”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Spread Horizontal”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

The network is spread to the right of the cursor position. No spread is possible before the first input (left edge of the screen).

The network is optimized when FBD elements are inserted.

Edit LD / Vertical Spread in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Spread Vertical”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Spread Vertical”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

A vertical spread is necessary before entering a further parallel line. The cursor must point to below the serial ladder diagram line. The network is spread above the cursor position.

A vertical spread corresponds to <Ctrl>+<V> or the <Ins> key outside the pull-down menu.

Edit LD / Normally Open Contact in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”N/O”
- ”Edit”, ”Block”, Block, <Return> ”Modify”, <Return>, ”N/O”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Symbol ⊣⊢
Reference character <O>

This function edits a normally open contact to the right of the cursor position:

a) in the serial ladder diagram line:
The output is inserted simultaneously in a new network when the function is selected.

b) in the parallel ladder diagram line:
A normally open contact can be edited as the first element of a parallel ladder diagram line with this function.
The cursor must point at least 3 lines below an element.

Edit LD / Normally Closed Contact in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”N/C”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”N/C”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Symbol -|/|-
Reference character <C>

This function edits a normally closed contact to the right of the cursor position:

a) in the serial ladder diagram line:
The output is inserted simultaneously in a new network when the function is selected.

b) in the parallel ladder diagram line:
A normally closed contact can be edited as the first element of a parallel ladder diagram line with this function.
The cursor must point at least 3 lines below an element.

Edit LD / Output in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Output”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Output”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Symbol -( )-
Reference character: <R>

This function edits an output:

a) An output can be edited in the new network before pressing the <Return> key (before the pulldown menu) with <Ctrl>+<R>.
The first normally opened contact is inserted simultaneously.

b) In an existing network, this function multiplies an output signal .
The cursor must point at least 3 lines below the previous output.

Edit LD, FBD / PB/FB/SFB Conditional

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”PB/FB/SFB Conditional”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”PB/FB/SFB Conditional”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

A block is called conditionally with this function. The block call thus depends on the result of a signal address.

e.g. PB4

I2.2 $\boxed{}$ PB 4 is only called if I 2.2 = 1

You must yourself enter the block name of the block to be called. The address of the condition must be entered at the left of the block.

FB: User FBs must be declared before the call (see function block)

SFB: Standard FBs can be called directly either with their SFB number or their SFB name (e.g. ”SFB101” or ”INV”). If at least one space is entered instead of the SFB name and confirmed with <Return>, the Standard Function Block library (STDFB library) is displayed.
You can scroll in the library with <PgDn> and <PgUp>.
The SFB of the library currently displayed is selected with <Return>.

PB: can be called (also before programming)

OB: cannot be called

Edit LD, FBD / PB/FB/SFB Unconditional

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”PB/FB/SFB Unconditional”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”PB/FB/SFB Unconditional”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

You unconditionally call a block with this function, i.e. it is executed in each scan. You must yourself enter the block name of the block to be called. The blocks are entered corresponding to the function PB/FB/SFB Conditional (see above).

Edit LD / FBD Elements in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”FBD Elements”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”FBD Elements”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

The elements can be called from the pulldown menu with their reference characters or by selecting the menu lines and <Return>.

The specifications for the LD editor and for ”Edit”, ”Block” are valid for inputting FBD elements.

Furthermore:
Further FBD elements can only be inserted to the right of the existing FBD elements. The network is optimized during the insertion.

FBD elements of the same type (at the same location) may be exchanged. The cursor must point to the ladder diagram line of the first input in this case.

The possible FBD elements are listed in the section Display Mode (function block diagram, as of page 107).

Edit LD / Erasing in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Erase”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Erase”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

This function erases elements of a network. The LD elements to the right of the cursor are erased in the LD editor. For FBD elements, the cursor must point to the ladder diagram line of the first input during erasing.

Erasing corresponds to <Ctrl>+<S> or the <Del> key outside the pulldown menu.

Edit LD / Parallel Line

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, Parallel Line
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, Parallel Line
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

These and the following functions are valid if you want to edit a second parallel ladder diagram line in the network in addition to the first one. Move the cursor at least 3 positions below a normally closed/normally open contact of the above ladder diagram line. The pulldown menu can be called with <Return>. (The menu is only opened if input is permitted at the cursor position.)

The menu is displayed until the end of the parallel line. The following functions can be selected:

Edit LD / Normally Open Contact in Parallel Line

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, Parallel Line, N/O Contact
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, Parallel Line, N/O Contact
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Symbol ⊣⊢
Reference character: <O>

This function inserts a normally open contact in a parallel line.

Edit LD / Normally Closed Contact in Parallel Line

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, Parallel Line, N/C Contact
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, Parallel Line, N/C Contact
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Symbol -|/|-
Reference character: <C>

This function inserts a normally closed contact in a parallel line.

Edit LD / Return of Parallel Line

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, Parallel Line, Return
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, Parallel Line, Return
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Symbol ——|
Reference character: <V>

The opened parallel ladder diagram line is closed at the cursor position.

Caution The symbol may only be inserted if the parallel line is correctly created. Otherwise the complete incorrect line is erased.

Edit LD / Continuation of Parallel Line

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, Parallel Line, Continuation
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, Parallel Line, Continuation
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Symbol ——
Reference character: <H>

The ladder diagram line is continued without contacts.

Function block diagram FBD

FBD Editor

- ”Edit”, ”Block”, Block, <Return>
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Note If you select an existing block in the required display mode, you can still process it at a later time. You enter the menu ”Edit Network” after pressing <Return>. Its structure is the same for all display modes. The common functions are described in IL.

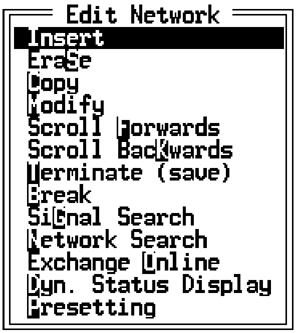
You are in the FBD editor, which will be described below.

You can edit in FBD

a) in the pulldown menu ”Edit Network” using the reference characters by selecting the menu line with the cursor keys and calling with <Return> or using the mouse,

b) with <Ctrl>+<reference character> from the higher editor level

a) The following functions are provided in the pulldown menu ”Edit Network”:



b) If you did not select the menu ”Edit Network”, you can execute the functions with <Ctrl>+<reference charcters>, e.g. <Ctrl>+<**M**> for ”**M**odify”. The following keys can also be used:

<PgDn>	Scroll to next network
<PgUp>	Scroll to previous network
<Esc>	Abort input without saving
<↑>	Shift screen up
<↓>	Shift screen down

If you press the <Return> key, you can also select the above-mentioned functions with the menu line of the pulldown menu ”Edit Network” appearing at the lower right of the screen.

When creating networks in FBD, note that elements can only be input at certain cursor positions.
You must select these positions with the cursor keys:

Insert:	to the right of the cursor; cursor must point to the first character of the ladder diagram address.
Delete:	to the right of the cursor; the cursor must point to the ladder diagram line of the input.
Delete FBD element:	Elements of the same type can be exchanged if the cursor points to the first input (ladder diagram line)
Insert FBD element:	to the left of the cursor; the cursor must point to the ladder diagram line of the output.

Automatic cursor positioning is possible for address input with the menu ”Presetting”.

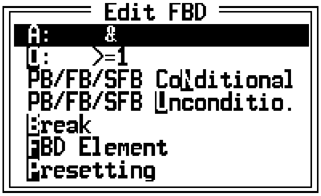
The address input begins with the first question mark.

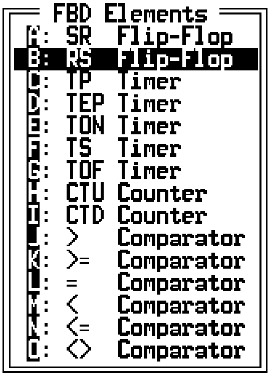
FBD Editor / Correction Mode

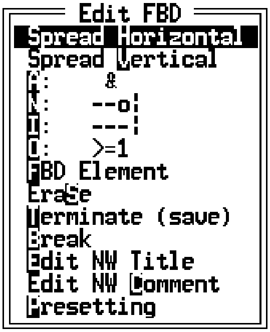
- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

You enter correction mode of the FBD editor from the menu ”Edit Network” (existing block) or from the menu ”Edit FBD” (new block, after entering the first element).

a) There are 2 pulldown menues ”Edit FBD” and one menu ”FBD Elements”, which can occur during editing:







b) You can execute the functions with <Ctrl>+<reference characters>, e.g. <Ctrl>+<**F**> for ”**F**BD Elements”. The following keys are also used:

<←>, <→>, <↑>, <↓>	Move cursor
<Ctrl>+< → >	Horizontal spread
<Del>	Delete element
<PgUp>	Shift image up line by line
<PgDn>	Shift image down line by line
<Ctrl>+<PgDn>	Shift image up page by page (scroll forwards)
<Ctrl>+<PgUp>	Shift image down page by page (scroll backwards)
<Letter>	Change contact address
<Esc>	Abort without saving

If you press the <Return> key, you can also execute the above-mentioned functions with the menu lines in the different pulldown menues ”Edit FBD”.

Edit FBD / AND-Block in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”AND”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”AND”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Symbol &
Reference character <A>

This function edits an AND block (with max. 46 inputs).
The network is optimized when the element is inserted.

Edit FBD / Negate Input in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Negate Input”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Negate Input”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Symbol
Reference charcter <N>

This function negates an input. Outputs cannot be negated. It is only possible to negate an existing input if the cursor points to the first character of the contact address.

A negated input is inserted if the cursor points directly to the left of the symbol.

The network is optimized when this element is inserted.

Edit FBD / Input in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Input”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Input”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Symbol
Reference character <I>

This function edits an input.

An additional input can be inserted if the cursor points directly to the left of the existing symbol.

Negated inputs can be overwritten.

The network is optimized when this element is inserted.

Edit FBD / OR-Block in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”OR”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”OR”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

Symbol >=1
Reference character <O>

This function edits an OR block (with max. 46 inputs).

The network is optimized when the element is inserted.

Edit FBD / Horizontal Spread in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Spread Horizontal”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Spread Horizontal”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

The network is spread horizontally to the right of the cursor position.
The network is optimized when FBD elements are inserted.

Horizontal spreading corresponds to <Ctrl>+<H> or <Ctrl>+< → > outside the pulldown menu.

Edit FBD / Vertical Spread in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Spread Vertical”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Spread Vertical”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

A vertical spread is necessary to insert further entries into an FBD symbol.

The cursor must point below the first input ladder diagram line of the FBD symbol.

The network is optimized during insertion.

Vertical spreading corresponds to <Ctrl>+<V> or the <Ins> key outside the pulldown menu.

Edit FBD / Erasing in Correction Mode

- ”Edit”, ”Block”, Block, <Return>, ”Insert”, <Return>, ”Erase”
- ”Edit”, ”Block”, Block, <Return>, ”Modify”, <Return>, ”Erase”
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”,..

This function erases elements from a network.

Erasing one element:	Cursor must point to the upper left corner of the block.
Erasing several, successive elements:	Cursor must point to the output of the chain of elements to be erased.

The network is optimized when this function is executed.

Erasing corresponds to <Ctrl>+<S> or the <Del> key outside the pulldown menu.

4.3.3 Symbols and Comments

- ”Edit”, ”Symbols and Comments”

This software function creates the SYM/COM block, which contains symbolic names, comments and initial values for signal addresses.

Note Suggestions for symbolic names, initial values and comments for the closed-loop control and POS data structures can be imported separately. The files ”REGELN.ASD” and ”POS.ASD” contain the relevant information (see page 299).

You enter the editor of the SYM/COM block by pressing the <Return> key.

SYM/COM Block

In order to show the relationship between an absolute address (input/ output, markers, etc.) and their technologic function, you can assign absolute addresses with symbolic names and comments. Furthermore, initial values can be assigned for each signal address.

The text of the symbolic names, comments and initial values are stored in the SYM/COM block under the current station name.

When programming, the symbolic names entered in this editor can be used instead of the absolute addresses after activating the symbols by setting the addressing to ”SYM”.

The SYM/COM block can be documented under ”Print”, ”Symbols and Comments”.

Standard values are preset for the system markers (bit, word, double word and floating point word).

SYM/COM Editor

The editor described here provides you with menus, key macros and the line editor. You can call further menues within the editors with <Ctrl>+<Return>.

Caution **Select the function "Replace Signals" in order to assign a symbol to another address (e.g. ANNA should now be assigned to pin I3.5 instead of I2.1). This is not possible by a modification at this location ! The program and the PLC always work with absolute addresses. If "SYM" is selected, the symbol is simply faded in for the user.**

Input Symbols, Comments

- "Edit", "Symbols and Comments", <Return>

Symbols are input with the SYM/COM editor:
After activation, the signal, symbol, initial value and comments table of the first inputs are displayed.

SYM/COM - Editor

Signal	Symbol	Init.-value	Comment
M1.1	RESET	-	"FOR THE 1ST CYCLE=1" AFTER PROGR.START
M1.2	PULSE_1	-	0,3125 HZ FLASHING PULSE
M1.3	PULSE_2	-	0,625 HZ FLASHING PULSE
M1.4	PULSE_3	-	1,25 HZ FLASHING PULSE
M1.5	PULSE_4	-	2,5 HZ FLASHING PULSE
M1.6	PULSE_5	-	5,0 HZ FLASHING PULSE
M1.7	ENC_BIT	-	ENCODER BIT : STANDERDISING OCCURED
M1.8	STARTUP	-	
M1.9		-	
M1.10	FIXED"0"	-	FIXED VALENCE "0"
M1.11	FIXED"1"	-	FIXED VALENCE "1"
M1.12	UKA_BIT1	-	JUMPER G INSERTED ON UKA
M1.13	UKA_BIT2	-	UKA SINGLE BIT 2 (CONSOLE ALLOCATION)
M1.14	VRUN	-	VLIST RUNNING
M1.15		-	
M1.16		-	

Line: 1 Column: 1 Type : Bit Read data: Yes
▸ overwrite ◂ <CTRL-ENTER> ≡ Commands

You can assign symbols, initial values and comments to the displayed addresses with special keys, the line editor and the key macros.

Symbols

A maximum of 8 characters may be entered for symbols.

The following characters may not be used for symbols:

- ä, ö, ü, Ä, Ö, Ü, ~, }, {, |
  these characters cannot be used in the block editors
- , (comma)
  is ignored when importing symbols and comments
- ; (semicolon)
  is converted into _ (underlining) when importing
- (space)
  is suppressed during importing (the subsequent characters are shifted left)

Initial Values

An initial value can be assigned to each symbol or absolute address. The values are written directly into signal memory of the PLC with the functions ”Program to PC*” and ”Initial Values to PC*”. Initial values can be transferred separately into the PLC after power failure. Initial values are valid for the first scan during the initial start and cold restart.

Special keys:

<↑>, <↓>, <←>, <→>	Move cursor
<Del>	Delete character undr cursor
<backspace>,< ⇐ >	Delete character to left
(tab)	Next input field
shift (tab)	Previous input field
<PgUp>	Previous page
<PgDn>	Next page
<Ctrl>+<PgUp>	Scroll line up
<Ctrl>+<PgDn>	Scroll line down
<Home>	Jump to upper screen edge
<End>	Jump to lower screen edge
<Ins>	Insert / Overwrite
<Esc>	Abort without saving

Only the input / output addresses defined in the equipment list under ”Number” are displayed.

The window is called with the following functions with <Ctrl>+<Return> :

```
Commands
Terminate (save)
Search Function
Search Operand
Erase Field
Copy Area
Erase Area
Relocate Area
Text Memory Editor
Search Text
Replace Text
Insert Comment Line
```

You can call the functions with <Ctrl>+<reference character> outside the menu.

Search Function

- ”Edit”, ”Symbols and Comments”, <Return>, <Ctrl>+<Return>, ”Search Function”

```
Search Function
You can search for signals or symbols.
Search for :
```

Enter the exact signal designator, e.g. for DIN M1.1, I2.18, MW1330,

Search Operand

- ”Edit”, ”Symbols and Comments”, <Return>, <Ctrl>+<Return>, ”Search Operand”

Operands can be searched after selecting from the following window. The first addresse of the selected operand type is set in the first line of the screen (you can only search for operand groups).

You open a selection window with all the possible operands by entering a space and <Return>. You select the operand with the cursor keys and call it with <Return>.

Erase Field

- ”Edit”, ”Symbols and Comments”, <Return>, <Ctrl>+<Return> ”Erase Field”

The input field to which the cursor pointed before the menu was called is erased. See also ”Erase Area”

Copy Area

- ”Edit”, ”Symbols and Comments”, <Return>, <Ctrl>+<Return> ”Copy Area”

You can also assign comments and initial values to other signals. The copied comments are maintained at the source.
A window appears on the screen when this function is selected.

Copy Area	
from signal :	first signal to be copied
thru signal :	last signal to be copied
to signal :	target address, first signal

The input may be either absolute or symbolic.

Note No symbols are copied at this location since otherwise double assignments could occur. The function ”Relocate Area” should be used to assign the symbols to other addresses.

Erase Area

- ”Edit”, ”Symbols and Comments”, <Return>, <Ctrl>+<Return>
  ”Erase Area”

You can erase symbols and comments in a block.
A window appears on the screen when this function is selected.

```
Erase Area
from signal  :
thru signal  :
```

first signal to be erased
last signal to be erased

The input may be either absolute or symbolic.

Relocate Area

- ”Edit”, ”Symbols and Comments”, <Return>, <Ctrl>+<Return>
  ”Relocate Area”

You can relocate symbols and comments as a block
A window appears on the screen when this function is selected.

```
Relocate Area
from signal  :
thru signal  :
to signal    :
```

first signal to be relocated
last signal to be relocated
target address, first signal

The input can be either absolute or symbolic. The text disappears from the source when this function has been executed (in contrast to copying).

Text Memory Editor

- ”Edit”, ”Symbols and Comments”, <Return>, <Ctrl>+<Return> ”Text Memory Editor”

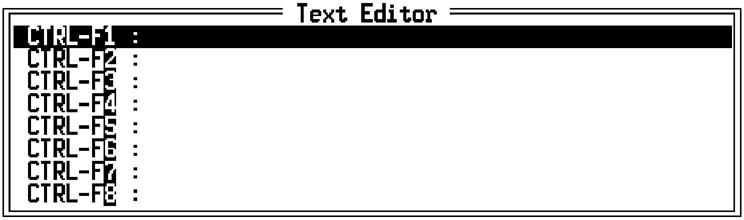
To avoid having to enter frequently recurring comments repeatedly, you can assign up to 40 characters of text to the function keys <F1> ... <F8>.

The assignment is valid until you leave the SYM/COM editor.

The text is entered at the current cursor position with <Ctrl>+<function key>.

Note Key macros (permanent memory) can also be used instead of the function keys.

After selecting the text memory, a pulldown menu appears into which you can enter text after selection with <Return>.



Search Text

- ”Edit”, ”Symbols and Comments”, <Return>, <Ctrl>+<Return>
  ”Search Text”

You can search for words, parts of words or parts of records in all the columns of the SYM/COM block. Signal addresses can also be defined.
A window appears on the screen when this function is selected.

```
==========Search Text==========
Search
Options:
G=Start at beginning of Block
U=Ignore small/capital letters
```

”G” and/or ”U” can be entered as options. The function can be repeated as often as required outside the pulldown menu with <Ctrl>+<A> .

Replace Text

- ”Edit”, ”Symbols and Comments”, <Return>, <Ctrl>+<Return> ”Replace Text”

You can replace words, parts of words or parts of records in all the columns of the SYM/COM block.
A window appears on the screen when this function is selected.

```
Replace text
Replace:
with:
Options:
G=Start at beginning of Block
U=Ignore small/capital letters
N=Without acknowledgement
```

”G” and/or ”U” and/or ”N” may be entered as options. If ”N” is entered, replacement is without acknowledgement request. The function can be repeated as often as required outside the pulldown menu with <Ctrl>+<R> .

Insert Comment Line

- ”Edit”, ”Symbols and Comments”, <Return>, <Ctrl>+<Return> ”Insert Comment Line”

An empty comment line of 62 characters width can be inserted at the cursor location with this function. Insertion is made above the current line.

4.3.4 Equipment List

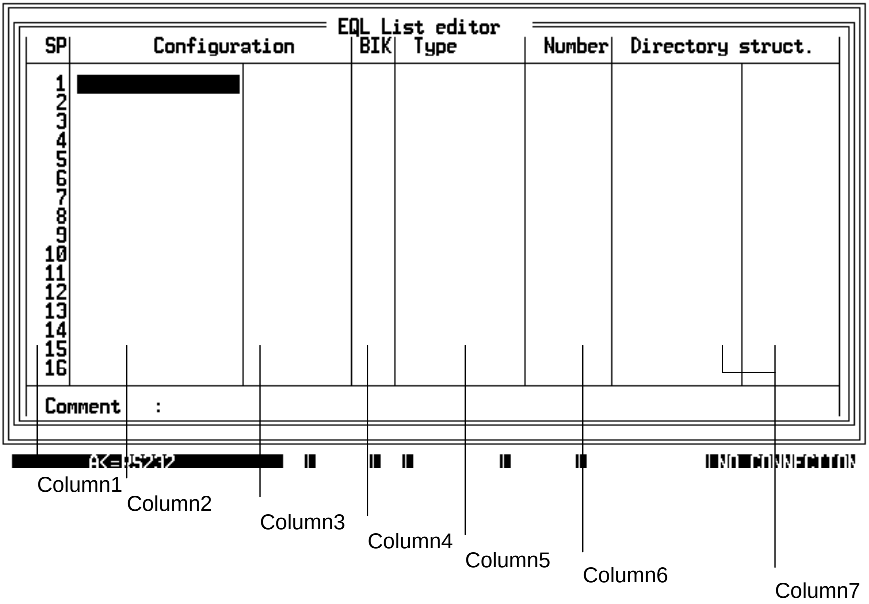
- "Edit", "Equipment List"

The equipment list contains the hardware equipment of the PLC to be configured.

☞ **Note** The referencing can be found in the relevant PLC user manual.

The equipment list must be edited before the blocks since a plausibility check is made. Input and output signals may only be entered in the user program if the corresponding module was first entered in the equipment list.

The following window appears when this function is selected



Subracks and I/O modules for rear or front connection are entered in the equipment list.

Caution **Enter only the Modnet 1/SFB nodes which are really to be driven. Otherwise timeout errors can occur when operating the station with Modnet 1/SFB networking (”AKF on BUS”).**

Column 1 (SP)
Column 1 contains the number of the slot reference (slot referencing is described in the relevant PLC user manual).

A maximum of 160 slot addresses is allowed. You must reserve two SP lines for modules with 8T width!

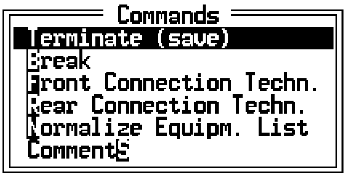
The UKA module is always on slot reference 1 for ALU types ALU 821, ALU 150 and ALU 286.

Subracks are put on the first slot reference. The corresponding slots are then assigned automatically.

Note The slot reference set with HW jumpers must agree with the number in ”SP”.

Column 2 (Configuration)
Rear-connection modules and front-connection subracks are entered in this column.

The following menu appears after <Ctrl>+<Return> :



Equipment List Editor / Terminate (save)

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Terminate”

The input is terminated and the alterations saved with this function.

Termination corresponds to the key <Ctrl>+<T> outside the pulldown menu.

Equipment List Editor / Break

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Break”

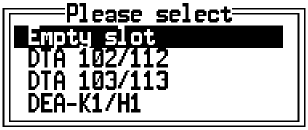
The input is aborted and the editor is left with this function.

Aborting corresponds to the key <Esc> or <Ctrl>+<B> outside the pulldown menu.

Equipment List Editor / Front Connection Techn.

- ”Edit”, ”Equipment List”, Column 2, <Ctrl>+<Return>, ”Front Connection Techn.”

At the first slot reference of the subrack to be equipped, <Return> is used to select one of the following:



Subrack with 4 I/O slot references
Subrack with 9 I/O slot references
Decentral I/O

You can only delete a subrack at its first slot reference in column 2. The input ”nempty slot” is used here.

You can call ”Front Connection Techn.” outside the pulldown menu with <Ctrl>+<F>.

Equipment List Editor / Rear Connection Techn.

- ”Edit”, ”Equipment List”, Column 2, <Ctrl>+<Return>, ”Rear Connection Techn.”

At the corresponding slot reference, <Return> is used to select one of the following:

Please select	
Empty slot	
32 Bit Input	Digital input 16 bit
16 Bit Input(A)	Digital input 16 bit, A line
16 Bit Input(E)	Digital input 16 bit, E line
32 Bit Output	Digital output 32 bit
16 Bit Output	Digital output 16 bit
DKU022	I/O bus monitoring
SES2	Interrupt-Input 16 bit
DPL011	Hexadecimal I/O
POS001/002/011	Intellig. function module positioning

The module is deleted with the input ”empty slot”.

You can call ”Rear Connection Techn. ” outside the pulldown menu with <Ctrl>+<R>.

Equipment List Editor / Normalize

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Normalize Equipm. List”

You completely delete the equipment list without leaving the editor with this function.

Normalization corresponds to the key <Ctrl>+<N> outside the pulldown menu.

Equipment List Editor / Enter Comments

- ”Edit”, ”Equipment List”, <Ctrl>+<Return>, ”Comments”

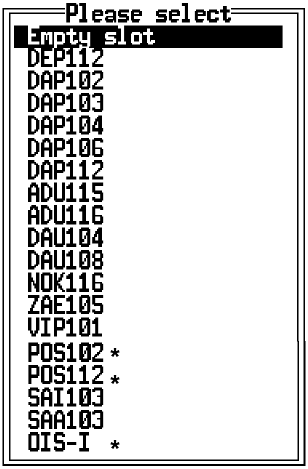
You enter a comment for a slot reference in the lowest line of the editor with this function. The comment is always valid for the line to which the cursor is pointing and it may have up to 40 characters.

You can enter a comment outside the pulldown menu after <Ctrl>+<S>.

Column 3 (Configuration)
Input is only possible here for front connection subracks.

The individually equipped I/O modules are entered (according to their slot reference) after <Return>.

The following modules are possible:



DEP112	Digital input 32 bits isolated
DAP102	Digital output 16 bits / Digital input 16 bits
DAP103	Digital output 16 bits / Digital input 16 bits
DAP104	Digital output 8 bits / Digital input 8 bits
DAP106	Digital output 16 bits
DAP112	Digital output 32 bits isolated
ADU115	Digital output 32 bits isolated
ADU116	Analog input 16 channels isolated
DAU104	Analog input / Analog output
DAU108	Analog output 8 channels isolated
NOK116	Intellig. function module CAM controller (16 cams)
ZAE105	Intellig. function module with 5 counters
VIP101	Intellig. function module operating/display technology
POS102*	Intellig. function module positioning, incremental
POS112*	Intellig. function module positioning, absolute
SAI103	Intellig. function module switching axis pos. incremental
SAA103	Intellig. function module switching axis pos. absolute
OIS–I*	Intellig. function module; Infos from AEG Weinheim

* the modules marked in this way can only be used with basic software version 6.0 (ALU 021, ALU 071)!

Column 4 (BIK)
This column contains the BIK number for front connection subracks. You can toggle between 1, 2 and 3 with <Return>.

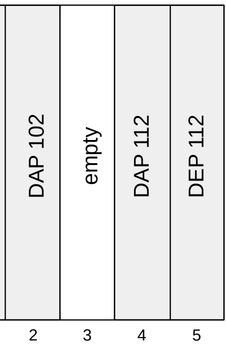
Column 5 (Type)
The basic data structures are displayed in this column. These are used to provide the RAM on the PLC for the intelligent function module (actual value/setpoint fields). You can look at the basic types under ”Edit”, ”Data Structures”.

Column 6 (Number)
The input/output addresses of the SYM/COM block and the user program refer to this number and not to the number of the first column (”SP”). The software is now independent of the slot references ; ”Assignment ” is possible within the equipment list.

It is therefore possible to use the same program with different hardware configurations. You adapt the number of the module to the user program (in this column). It is not necessary to adapt the user program to the slot references (SP).

The following figure shows an example. The same user program can be used for both hardware configurations with Q9.1 to Q9.32, I10.1 to I10.32, Q11.1 to Q11.16 and I11.17 to I11.32.

☞ **Note** When the equipment list is read out of the PLC, the logical slot references are only maintained if you execute the function ”Load”, ”Read Out PC*”.



st editor	
Type	Number
],I	4
]	2
	3

First of all the software automatically enters the ”SP” number in this column as well. The number can be altered here.

Note Only certain number ranges are permitted for intelligent function modules. (The software shows if there is incorrect input). You then enter the number in the user program at parameter ”TN” or ”SP” of the FBs or specific SFBs.

The following numbers are permitted:

Module	**Possible number SP**	**TN**	**Module**	**Possible number SP**	**TN**
Rear connection			**Front connection**		
32 bits I	1 ... 160	1 ... 160	DEP 112	1 ... 160	1 ... 160
16 bits I (Q)	1 ... 160	1 ... 160	DAP 102	1 ... 160	1 ... 160
16 bits I (I)	1 ... 160	1 ... 160	DAP 103	1 ... 160	1 ... 160
32 bits (Q)	1 ... 160	1 ... 160	DAP 104	1 ... 160	1 ... 160
16 bits (Q)	1 ... 160	1 ... 160	DAP 106	1 ... 160	1 ... 160
DKU 022	–	–	DAP 112	1 ... 160	1 ... 160
SES2	1 ... 160	1 ... 160	ADU 115	–	–
DPL 011	1 ... 160	1 ... 160	ADU 116	–	–
POS001/002/011	1 ... 160	1 ... 160	DAU 104	–	–
			DAU 108	.	–
			NOK 116	1 ... 160	1 ... 99*
			ZAE 105	1 ... 160	1 ... 99*
			VIP 101	1 ... 160	1 ... 10*
			POS102/POS112	1 ... 160	1 ... 31*
			SAI103/SAA103	1 ... 160	1 ... 99*

Column 7 (Directory Structure)
This column is only used for display and is meaningless for the AKF user.

Example:
Mixed equipment rear connection/front connection

An example of equipment with front and rear connection in A350 is shown here. The following are to be connected:

- ❒ rear connection : 3 POS 011 (width 8T), 4 SES 002 (width 4T)
- ❒ front connection : 2 DAP 102, 1 DEP 112 (width 8T each) in one subrack DTA 150.

☞ **Note** The equipment list entry of the front connection subrack is only according to width. DTA 102/112 is always entered for subracks with 4 I/O slots. DTA 103/113 is always entered for subracks with 9 I/O slots.

EQL List editor

SP	Configuration		BIK	Type	Number	Directory struct.	
1	UKA Module						
2	POS001/002/011			Q,I	2		
3							
4	POS001/002/011			Q,I	4		
5							
6	POS001/002/011			Q,I	6		
7							
8	SES2			I	8		
9	SES2			I	9		
10	SES2			I	10		
11	SES2			I	11		
12							
13	DTA 102/112	DAP102	1	Q,I	13		
14	*	DAP102	1	Q,I	14		
15	*	DEP112	1	I	15		
16	*	-	1				

Comment : Digital Input 32 bit

Caution You may not equip the slot references more than once in a mixed structure.

4.3.5 Title Block

- "Edit", "Title Block"

The following functions are provided in the pulldown menu "Title Block":

```
Edit Title Block
Start Entry
Title Block Name   TITLE.DOC
Title Block Width  132
Title Block Height 6
```

The input is made with a title block editor.

Start Entry

- "Edit", "Title Block", "Start Entry"

After definition of the title block name, width and height, input starts by pressing <Return>. This title block can then be used in all print operations. Output is always at the end of a print page. If the title block file does not exist, it will be created automatically with the parameters defined in the pulldown menu (name, width, height).

Title Block Editor

- "Edit", "Title Block", "Start Entry", <Return>

You enter the title block for the station documentation lists to be output to the printer, file or screen with this editor.

You can edit any text in the displayed frame with the following special keys and the typewriter keyboard.

Special keys:

<Del>	Delete character under cursor
<backspace>,< ⇐ >	Delete character to left
<↑>, <↓>, <←>, <→>	Move cursor
<Return>	Terminate line input
backtab	Previous input field
<PgUp>	Shift image to left
<PgDn>	Shift image to right
<Ins>	Insert / overwrite
<Ctrl>+<B>	Save + terminate editing

Graphic Characters:
You can also use graphic characters when creating a new title block. These are generated using the keyboard with the <Alt> key and the keys of the right number block:

Switch on <NumLock>. Press the <Alt> key and keep it pressed down. Enter a 3-digit number with the keys of the right number block and release the <Alt> key again. The corresponding graphic character appears at the current cursor position on the screen.

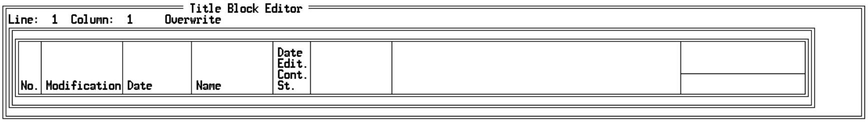
The following graphic characters are available (the 3-digit number at the left, the corresponding character at the right)

179 │	180 ┤	181 ╡	182 ╢	183 ╖
184 ╕	185 ╣	186 ║	187 ╗	188 ╝
189 ╜	190 ╛	191 ┐	192 └	193 ┴
194 ┬	195 ├	196 ─	197 ┼	198 ╞
199 ╟	200 ╚	201 ╔	202 ╩	203 ╦
204 ╠	205 ═	206 ╬	207 ╧	208 ╨
209 ╤	210 ╥	211 ╙	212 ╘	213 ╒
214 ╓	215 ╫	216 ╪	217 ┘	218 ┌

Figure 19 Available Graphic Characters

You can also use a standard title block file instead of setting up your own title block file. You need only enter the current specifications in the standard title block file.

The standard title block has the following appearance:



Title Block Name

- ”Edit”, ”Title Block”, ”Title Block Name”

Enter the name of the title block file to be processed here.

The standard title block file is called ”TITLE.DOC”. The extension ”.DOC” is normally specified for title block file names.

Title Block Width

- ”Edit”, ”Title Block”, ”Title Block Width”

Enter the required title block width here. Files which were already created cannot be altered afterwards.

64-132 characters are permitted.

Title Block Height

- ”Edit”, ”Title Block”, ”Title Block Height”

Enter the required number of lines in your title block here (valid values 1-12). Files which were already created cannot be altered afterwards.

4.3.6 Replace Signals

- "Edit", "Replace Signals"

You can exchange signal addresses with others using this function.
A block list must be entered here. With "*" an exchange is done in all the blocks of the current station.

You can select and call the following functions in this pulldown menu:

```
═══════════════ Replacement of Signals ═══════════════
 Start Replacement
 Old Signal    M 1.1
 New Signal    M 1.1
 Block List:   PB
 Load Block List
 Store Block List
```

Make sure that the documentation is adapted to the new situation with the SYM/ COM block.

If you want to exchange two contacts A and B, proceed as follows:

Step 1 "Old Signal" → A, "New Signal" → Auxiliary contact, Start replace

Step 2 "Old Signal" → B, "New Signal" → A, Start replace

Step 3 "Old Signal" → Auxiliary contact, "New Signal" → B, Start replace

The auxiliary contact is only required for this exchange and must be of the same type as contacts A and B. This procedure is necssary to prevent the loss of information because if one would replace contact A with contact B and then contact B with contact A, contact A would be lost during the first use of the replace function.

Start Replacement

- "Edit", "Replace Signals", "Start Replacement"

The replace is started when all the necessary data has been specified (see following pulldown menu lines).

Old or New Signal

- "Edit", "Replace Signals", "Old or New Signal"

Enter the required hardware address here (AEG, DIN: I, Q, M, MW, MD, MF etc.). The signal type for the old and new signals must agree.

Block List

- "Edit", "Replace Signals", "Block List"
- "Online", "Dyn. Status Display", "Single Shot Processing", "Block List"
- "Online", "Dyn. Status Display", "Triggered Recording", "Block List"
- "Load", "Read Out PC*", "Block List"
- "Load", "Compare", "Blocks with PC*", "Block List"
- "Print": "Program Log", "Block List"
  "Cross-Reference List", "Block List"
  "Signal Occupancy List", "Block List"
- "Special", "Import", "Blocks", "From Export File", "Block List"
- "Special", "Import", "Blocks", "From Station", "Block List"
- "Special", "Import", "Blocks", "From DOS File", "Block List"
- "Special", "Export", "Blocks", "To Export File", "Block List"
- "Special", "Export", "Blocks", "To DOS File", "Block List"

Enter a list of all the blocks to be processed here.
The list may have max. 200 characters.

*	→	Processing of all the blocks existing in the station. "*" is the default value.
Blank + Return	→	a selection window of all the blocks appears
OB, PB, FB	→	e.g. OB1, PB1, FB55, ...
PBxxx-yyy	→	e.g. PB8-19
FB10	→	only FB10

In a list, the definitions must be separated by commas: e.g. PB1-19, PB23-24, FB5-8, FB13-28

Existing entries can be overwritten or deleted with the <blank>-key and then re-defined.

This list can be stored under a name with "Store Block List" and then be loaded again at any location as often as required.

Load Block List

- ”Edit”, ”Replace Signals”, ”Load Block List”
- ”Load”, ”Read Out PC*”, ”Load Block List”
- ”Load”, ”Compare”, ”Blocks with PC*”, ”Load Block List”
- ”Print”: ”Program Log”, ”Load Block List”
  ”Cross-Reference List”, ”Load Block List”
  ”Signal Occupancy List”, ”Load Block List”

The list created under ”Block List” and stored under ”Store Block List” can be loaded again with this function as required.

Store Block List

- ”Edit”, ”Replace Signals”, ”Store Block List”
- ”Load”, ”Read Out PC*”, ”Store Block List”
- ”Load”, ”Compare”, ”Blocks with PC*”, ”Store Block List”
- ”Print”: ”Program Log”, ”Store Block List”
  ”Cross-Reference List”, ”Store Block List”
  ”Signal Occupancy List”, ”Store Block List”

The list entered under ”Block List” can be stored under a name of your choice. You can call it again as needed with ”Load Block List”.

4.3.7 Compile (Blocks)

”Edit”, ”Compile (Blocks)”

The blocks which were created with older versions of the Dolog AKF software (OB, PB, FB) are automatically converted with this function so that they can be executed in the new version (station data base). You can select whether or not you want to delete the old blocks.

The following menu appears in this function:

Recommendation: The old blocks should first be archived and ”Delete Old Blocks” to ”on”.

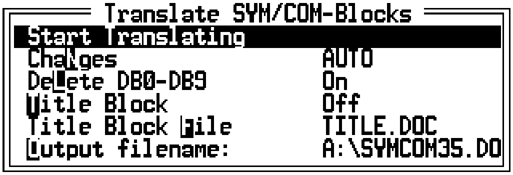
Expert If you want to use your own SFBs with numbers greater than 500, you must insert your SFBs in the library with LIBGEN.EXE (last software diskette). Use the newest version of ”LIBGEN.EXE”.

4.3.8 Translate (SYM/COM Blocks)

”Edit”, ”Translate (SYM/COM Blocks)”

SYM/COM blocks (DB0 ... BB9) of earlier Dolog AKF software versions are converted into the current version (data base) with this function.

The following functions are provided in this pulldown menu:



Start Translation

- ”Edit”, ”Translate (SYM/COM Blocks)”, ”Start Translating”

Note A correct translation requires a complete equipment list or an executed ”Translate (Blocks)”.
The function is time-intensive for long DBs.

The translation is started with this function after input of the settings. The whole station is searched for ”old” SYM/COM blocks (DB0-DB9). The contents of these blocks (symbols, comments, initial values) are transferred to the SYM/COM block (data base).

Changes

- "Edit", "Translate (SYM/COM Blocks)", "Changes"

You can select manual or automatic operation here.

Caution Since this function could combine several SYM/COM blocks to one SYM/COM block (data base), double assignments of symbols can occur.

In this case you can correct symbols during the transfer when in "manual" operating mode.

In "automatic" operating mode, doubly assigned symbols are not transferred to the data base. You can switch over the operating mode at any time with the key <↓> during the data transfer.

Delete DB0-DB9

- "Edit", "Translate (SYM/COM Blocks)", "Delete DB0-DB9"

If you enter "on" here (default value), all the old DBs (DB0-DB9) are deleted after the transfer to the SYM/COM block (data base). If you enter "off", the old DBs are retained.

Caution The old DBs are no longer available if you enter "on". For security reasons, make a copy of the old DBs before deleting.

Title Block

- ”Edit”, ”Translate (SYM/COM Blocks)”, ”Title Block”
- ”Load”, ”Compare”, ”Blocks with PC*”, ”Title Block”
- ”Load”, ”Compare”, ”Program with PC*”, ”Title Block”
- ”Print”: ”Overview”; ”Program Log”; ”Symbols and Comments”; ”Equipment List”; ”Cross-Reference List”; ”Signal Occupancy List”; ”Station Setup”; ”Signal Memory Contents”; ”Data Structures”, ”Title Block”
- ”Special”, ”Directory”, ”AKF Blocks” ”Title Block”
- ”Special”, ”Directory”, ”DOS Files” ”Title Block”
- ”Special”, ”System Information”, ”Title Block”
- ”Setup”, ”Station”, ALU 0xx, ”First Use of PC*”, ”BSW Configuration (BSW>=V6.0)”, Module, ”Print”, ”Title Block”

You can also output the title block which you generated in the menu option ”Edit”, ”Title Block” or which exists standardly under the name ”TITLE.DOC” and which was copied under the station.

You can toggle between ”on and ”off”.

The entry ”on” means that the title block whose file name is entered under the menu option ”Title Block File” appears on the printout. For the entry ”on”, the output is rejected with an error message if the required title block does not exist at the station.

Title Block File

- ”Edit”, ”Translate (SYM/COM Blocks)”, ”Title Block File”
- ”Load”, ”Compare”, ”Blocks with PC*”, ”Title Block File”
- ”Load”, ”Compare”, ”Program with PC*”, ”Title Block File”
- ”Print”: ”Overview”; ”Program Log”; ”Symbols and Comments”; ”Equipment List”; ”Cross-Reference List”; ”Signal Occupancy List”; ”Station Setup”; ”Signal Memory Contents”; ”Data Structures”, ”Title Block File”
- ”Special”, ”Directory”, ”AKF Blocks”, ”Title Block File”
- ”Special”, ”Directory”, ”DOS Files”, ”Title Block File”
- ”Special”, ”System Information”, ”Title Block File”
- ”Setup”, ”Station”, ALU 0xx, ”First Use of PC*”, ”BSW Configuration (BSW>=V6.0)”, Module, ”Print”, ”Title Block File”

Enter the name of a title block file existing under the station here if you defined ”on” in the menu option ”Title Block”. Otherwise the message ”Title block file not found. Press any key.” appears when the print is started.

In this case you must either enter the name of another title block file existing under the station or copy the corresponding title block file which was already created under this station in the menu option ”Special”, ”Copy Files”, ”DOS Files”. You can enter the file name with the line editor.

Output Unit

- ”Edit”, ”Translate (SYM/COM Blocks)”, ”Output Unit”
- ”Load”, ”Compare”, ”Blocks with PC*”, ”Output Unit”
- ”Load”, ”Compare”, ”Program with PC*”, ”Output Unit”
- ”Print”: ”Overview”; ”Program Log”; ”Symbols and Comments”; ”Equipment List”; ”Cross-Reference List”; ”Signal Occupancy List”; ”Command File”; ”Station Setup”; ”Signal Memory Contents”; ”Data Structures”, ”Output Unit”
- ”Special”, ”Directory”, ”AKF Blocks” ”Output Unit”
- ”Special”, ”Directory”, ”DOS Files” ”Output Unit”
- ”Special”, ”System Information”, ”Output Unit”
- ”Setup”, ”Station”, ALU 0xx, ”First Use of PC*”, ”BSW Configuration (BSW>=V6.0)”, Module, ”Print”, ”Output Unit”

The following are provided as output unit:

Output Unit	
Monitor	Output pagewise on screen
Printer	Output continuous on printer
File:	Output in a file (after <Return> you can enter a file name)

4.3.9 DOLOG-SFB Address List

- ”Edit”, ”DOLOG-SFB Address List”

Note This function can only be selected if the following condition is satisfied:
in ”Setup”, ”Station”, ”ALU Type” entries ALU 150, ALU 286, ALU 011, ALU 061 or ALU 821 (only BSW < version 6.0 possible)

Note For the case
”Setup”, ”Station”, ”ALU Type” ALU 021 or ALU 071, you configure the loadable basic software version 6.0 (”Setup”, ”Station”, ”First Use of PC*”, ”Configure BSW”) instead of the address list.

One of the following two standard address lists is generated in your station when this function is called.

❐ Standard Setting for ALU 150, ALU 011, ALU 286, ALU 061 (BSW < V6.0)

```
═══════════════════════ Address list editor ═══════════════════════
DOLOG-SFB Address list for ALU          (150,286,011,061)
                   with BSW version     (V5.x)
   G1  Basic software [MWVBSY]          Segment    (19)
                                        Block      (3)
   G2  Basic software [A48ERG]          Segment    (21)
                                        Block      (3)
   B1  [Reserve]                        Segment    (  )
                                        Block      ( )
   B2   Sequence control system /       Segment    (17)
        Mass flow                       Block      (1)
   B3   Front connection technique      Segment    (24)
                                        Block      (4)
   B4   Text processing                 Segment    (20)
                                        Block      (1)
   B5   Closed loop control             Segment    (18)
        Floating-point processing       Block      (1)
   B6   Positioning                     Segment    (17)
                                        Block      (2)
   B7   DOZ/DBS                         Segment    (17)
                                        Block      (4)
   B8   Backup-Controler                Segment    (19)
                                        Block      (1)
   B9   NOK                             Segment    (17)
                                        Block      (4)
   B10  ZAE                             Segment    (26)
                                        Block      (2)
#  Bn   Application program             Segment    (  )
                                        Block      ( )
+       Application block               Number     (   )
                                        Offset     (    )

────────────────────────────────────────────────────────────────────
Insert: <ins>, Delete: <del>, End/Abort: <esc>
```

❐ Standard Setting for ALU 821 (BSW V13.3)

```
                          Address list editor
DOLOG-SFB Address list for ALU      (821)
                    with BSW version (V13.3)
   B1  [Reserve]                     Segment    (  )
                                     Block      ( )
   B2   Sequence control system /    Segment    (25)
        Mass flow                    Block      (1)
   B3   Front connection technique   Segment    (24)
                                     Block      (4)
   B4   Text processing              Segment    (20)
                                     Block      (1)
   B5   Closed loop control          Segment    (18)
        Floating-point processing    Block      (1)
   B6   Positioning                  Segment    (25)
                                     Block      (2)
   B7   DOZ/DBS                      Segment    (25)
                                     Block      (4)
   B8   Backup-Controler             Segment    (19)
                                     Block      (1)
#  Bn   Application program          Segment    (  )
                                     Block      ( )
+       Application block            Number     (    )
                                     Offset     (     )

Insert: <ins>, Delete: <del>, End/Abort: <esc>
```

Note After entry or change, the program must be linked again and transmitted to the PC* (if this already happened).

Please check whether the basic software version of your PC* agrees with the specifications in the address list. If necessary, adjust your address list.

The list contains the default values for the segment and block numbers of the Dolog block packages as planned in the equipment recommendation for the A350/A500 in the documentation.

The block packages of the A500 can also be inserted and run at other address areas on ALU 821. Configurations other than the standard one must be altered accordingly in the address list. Make sure that the segment specifications are two-digit, i.e. segment numbers < 10 must contain a zero.

The indicator automatically moves to the next bracket. Furthermore, it can be moved freely in a vertical direction using the cursor keys.

The lines marked with ”#” and ”+” are intended for user packages.

You can extend the entries yourself. Pressing the <Ins> key duplicates either the ”user package” or ”user block” field, depending on the cursor position.

4.3.10 Command File

- ”Edit”, ”Command File”

The command file is used to create a complete system documentation. The required commands are first entered under ”Edit”. The file is processed under ”Print”.

Edit Command File

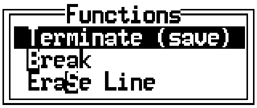
- ”Edit”, ”Command File”, ”Start Entry”

The file specified under ”Command File Name” is created or altered.

Command	Presetting	Meaning
$OVER	-	Output program survey
$PROT	-	Output program log
$SYMCOM	-	Output symbols / comments
$EQL	-	Output equipment list
$CRL	-	Output cross-reference list
$OCL	-	Output signal allocation list
$SETUP	-	Output Setup Station
$END	-	End of command file
$#xx	-	Control character printer
$TOF/$TON	-	Start/end of comment
$File=name	-	Output any ASCII file
$LD/$FBD/$IL	$FBP	Set output mode
$SYM/$NOSYM	$SYM	Symbols comments on/off
$LCRL/$NOLCRL	$LCRL	Local cross-reference list on/off
$Title=name	TITLE.DOC	Name title block file
$BL=xxxx	*	Block list
$SL=xxxx	*	Signal list
$LBL=file		Load block list from file
$LSL=file		Load signal list from file
$LOCAL/$GLOBAL	$LOCAL	Set cross-reference/assignment mode
$Station=xxxx	current station	Set station name

A standard command file is provided with ”DOK.KMD” (presetting).

The following window appears with <Ctrl>+<Return>:



The following functions are provided with the special keys:

<↑>, <↓>, <←>, <→>	Move cursor
<PgUp>	Scroll backwards
<PgDn>	Scroll forwards
<Ctrl>+<S>	Cursor left
<Ctrl>+<D>	Cursor right
<Ins>	Insert / overwrite
<Del> , <Ctrl>+<G>	Delete character under cursor
<backspace>	Delete character before cursor
<Ctrl>+<Y>	Delete line
<Home>	Cursor to start of line
<End>	Cursor to end of line
<Ctrl>+<T>	Terminate input with save
<Esc> , <Ctrl>+<B>	Abort input without save

Command File Name

- ”Edit”, ”Command File”, ”Command File Name”
- ”Print”, ”Command File”, ”Command File Name”

The name of the command file which you want to edit or print is specified here. It is preset to the standard file.

4.3.11 Data Structures

- "Edit", "Data Structures"

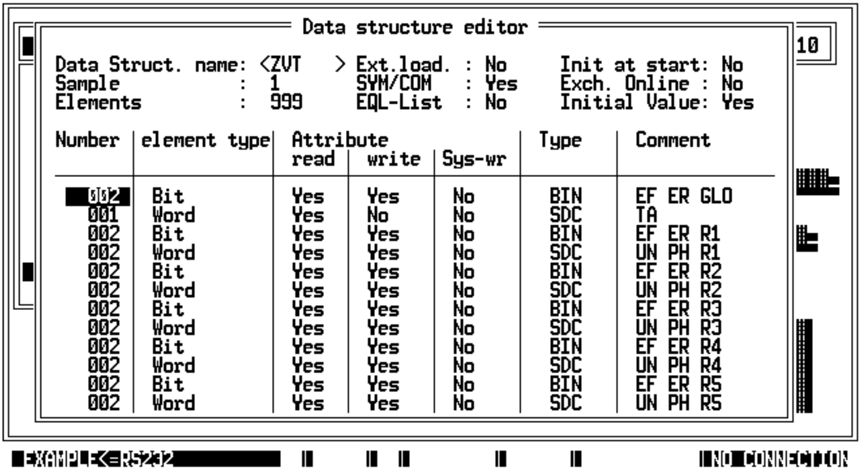
There are two types of data structures (general information: see page 500):

- ❐ data structures defined by AEG which cannot be altered (see also chapter 2.10)
- ❐ new data structures defined by the user

This editor is used to display the AEG data structures and to create and alter user data structures.

The line "Data Structure Name" appears after selection of this function. You can display a selection window with all existing data structures by entering a blank and <Return>. Select the required name with <Return>.

The following figure shows the data structure ZVT (time management table for closed-loop control; this structure was provided by AEG and cannot be modified) as an example.

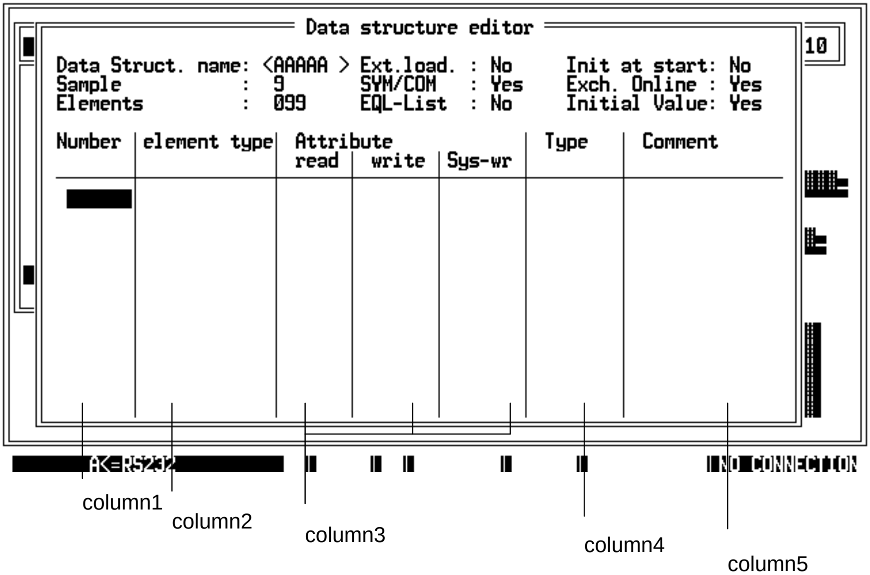


☞ **Note** The functions described below are used to create and modify user data structures.

Relationship between Length of Name, Number of Samples/Elements

There is a relationship between the possible name length, the possible number of samples and the resulting maximum elements (see also chapter 2.5.1.1). The name contains at least 4 alphanumeric characters (A...Z).

In order to create a new data structure, enter a new name in the line ”Data Structure Name” after selecting the function. The number of samples is then requested. You define how often a data structure may be used in a station with ”Sample”.



A description of the columns is followed by an explanation of the editor functions.

Column 1 (Number)

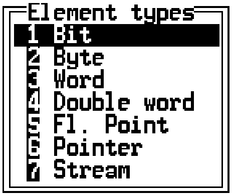
The number of related elements of the same type is defined in this column. These elements then occupy consecutive addresses in the SYM/COM block. The sum of the numbers in ”Number” is limited by the ”Maximum Elements” in the editor title.

The column is activated (cursor jumps to left) or the entered number is accepted with <Return>.

Column 2 (Element Type)

A data structure may comprise several different elements of a fixed element type. These element types are predefined.

You can select or accept one of the following element types in the second column after pressing <Return>:



Size	Values
1 Byte	0/1, binary
1 Byte	0 bis 255
2 Byte	-32 768 to +32 767
4 Byte	-2 147 483 648 to +2 147 483 647
4 Byte	± 1.1755 E-38 to 3.60282 E38
4 Byte*	
set and defined by software (internally)	

* with ”Pointer”, a segment can be specified with offset

Data structures may contain any combination of element types from bit to pointer or may contain only a stream.

Column 3 (Attribute)

Attributes can be assigned to the elements.

In the columns ”write” and ”read”, you can toggle between ”yes” and ”no” with <Return>.

This column is divided into:

- write = ”yes” Contents of element can be described by the user (e.g. in IL assigned with ”=”)
- read = ”yes” Contents of element can be read by user (e.g. in IL with ”LD”, comparable to constants)
- Sys-wr = ”yes” ”System write”, only used for display; the data structure is stored on RAM (in addition if user program is stored on EPROM)

The combinations
”write” = ”yes”, ”read” = ”no” (contents of element can only be written in user program)
”write” = ”no”, ”read” = ”no” (contents of element can be neither written nor read)
make no sense.

☞ **Note** If ”write” or ”sys-write” are ”yes” for an element, a copy of the whole data structure is stored on RAM (i.e. double storage requirements for each data structure).

Column 4 (Display Mode)

The display mode of the element is specified in this column:

Display mode	Used for element type
BIN (Bit pattern)	Bit
DEC (decimal)	Byte
SDC (± decimal)	Word, Double word
FLP (floating point format)	Floating word
HEX (hexadecimal)	Pointer, Stream

Column 5 (Comments)

You can assign up to 9 characters of comment to the element group in this column.

Editor Functions

You call the following pulldown menu with <Ctrl>+<Return>:

```
Actions
Data Structure Name
Insert Line
Delete Line
Load Data Structure
Delete Data Structure
Store Data Structure
Copy Data Structure
Terminate (save)
Abort
Change Presetting
```

Data Structure Name

- "Edit", "Data Structures", "Data Structure Name"

You can only change the name of the data structure with the data structure name option. The new data structure might have to be reloaded.

The alterations are made in the following menu:

```
=====Data structure name=====
 Data Struct. Name  AAAAA
 Samples            9
 Enter
```

Terminate the menu with "Enter" to save the alterations.

You can abort the editor with the <Esc> key, the alterations are lost.

Insert Line

- "Edit", "Data Structures", "Insert Line"

This function inserts a new line in the data structure at the current cursor position. The line is preset to a standard value. The function can only be used if the data structure or elements contained in it were not yet used in the program at that moment.

Delete Line

- "Edit", "Data Structures", "Delete Line"

This function deletes the line at the current cursor position in the data structure. This function can only be used if the data structure or elements contained in it were not yet used in the program at that moment.

Load Data Structure

- ”Edit”, ”Data Structures”, ”Load Data Structure”

The existing, stored data structure with the name contained in the editor title is loaded from the data base into the editor (corresponds to initial state).

If a new data structure was created and not yet stored, the data entered in the window are deleted with this function.

You can abort the editor input with the <Esc> key, in which case the alterations are lost.

Delete Data Structure

- ”Edit”, ”Data Structures”, ”Delete Data Structure”

The existing, stored data structure with the data structure name contained in the editor title is deleted in the data base. The window with the basic settings already entered (data structure name/sample) is retained. The function can only be used if the data structure or elements contained in it were not yet used in the program at that moment.

You can abort the editor input with the <Esc> key, in which case the alterations are lost.

Store Data Structure

- ”Edit”, ”Data Structures”, ”Store Data Structure”

The data structure specified with the data structure name and sample is stored in the data base. If a data structure exists with the same name, it is overwritten. The function can only be used if the data structure or the elements contained in it were not yet used in the program at the moment.

You can abort the editor input with the <Esc> key, in which case the alterations are lost.

Copy Data Structure

- ”Edit”, ”Data Structures”, ”Copy Data Structure”

The data structure in the editor is stored identically to a newly entered data structure name. The data structure name can also be accepted when the function is executed.

```
Copy data structure
Data Struct. Name  AAAAA
Samples           9
Start copy
```

You can abort the editor input with the <Esc> key, in which case the alterations are lost.

Terminate (save)

- ”Edit”, ”Data Structures”, ”Terminate (save)”

The editor is terminated and the processed data structure is stored in the data base.

You can also terminate outside the pulldown menu with the <F2> key.

Abort

- ”Edit”, ”Data Structures”, ”Abort”

The data structure in the editor is not stored if there is an abort. You leave the editor, just as with the <Esc> key.

Change Presetting

- "Edit", "Data Structures", "Change Presetting"

You can assign certain attributes to your user data structure with this function. You can toggle between "yes" and "no" in the following menu :

```
Presettings
Enter
Display in SYM/COM              :    Yes      yes/no
Online exchangeable             :    Yes      yes/no
Initial values exist            :    Yes      yes/no
External data structure         :    No       yes/no
Initialisation at PC* start:         No       yes/no
```

You can enter:

- Data structure with display in the SYM/COM editor
- Data structure is online exchangeable
- Data structure can be assigned initial values in the SYM/COM editor (see also page 300)
- Data structure is imported (no AKF35 source)
- Initial values are set when the program is started in the PLC ("Start PC*", Terminal Mode or Auto-Start). (Presetting "no") This is of no importance for the initial program start after the load since generally all initial values are set at this time.

This is displayed at the upper right side of the data structure editor.

Note AEG data structures cannot be altered.

4.3.12 Closed-Loop Control

- ”Edit”, ”Closed loop control”

Note General information about closed-loop control with AKF35 (runtime system etc.) can be found in the documentation ”A350/A500, Regeln mit Dolog AKF, Benutzerhandbuch, A91V.12-271963”[4)]

The runtime system ensures that different sampling times exist for the different closed-loop functions. A program under time control is set up and managed here with the SFB390, O-REG. The SFB O-REG works together with the ZVT (time management table) data structure. The OB linked to the O-REG block (e.g. OB2. called OBi below) is called under interrupt control with the basic sampling time t0_CLC entered in the time management table.

Call Marker
Three markers are processed in the ZVT for each control loop. These depend on the reduction and phase angle as well as the enable and reset input. The markers are used for the conditional call of PBs within the OBi. The corresponding control loop or its initialization is configured within the called PB.

The three markers are processed by the system (Dolog AKF) before each call of the OBi:

- START#xx single run before constant processing of the control loop
  The marker START#xx is used for the first initialization of the control loop xx. The user can define here what is to happen before the constant processing of the control loop xx.

- LAUF#xx constant processing of the control loop
  The marker LAUF#xx is used for the current processing of the control loop xx. It is set whenever the sampling time configured for the control loop has expired (e.g. once per second).

4) in german language

- HALT#xx single run after constant processing of the control loop
  The marker HALT#xx can be used for a final processing if the control loop xx is no longer to be processed.

☞ **Note** Symbolic names, initial values and comments for the closed-loop control data structures must be imported separately. The file ”REGELN.ASD” contains the corresponding information.

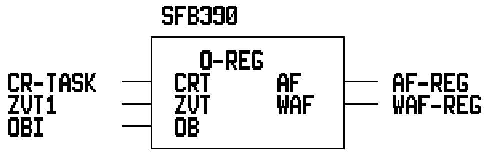
In addition to the call markers of the individual control loops, there are two call markers for the whole runtime system:

- STARTCLC single run before constant processing of all control loops
  The marker STARTCLC can be used for the first initialization of the complete closed-loop control. It is set once before any control loop can be called with the marker LAUF#xx.

- HALTCLC single run after constant processing of all control loops
  The marker HALTCLC can be used for a final processing (initialization) after the execution of all control loops. It is set once after the processing of all control loops if the runtime system is inhibited.

Example: Setting up Closed-Loop Control

Network xx in the OB1:



Example: Use of the Call Markers

OB2	START#1	PB1	Initialization before constant operation of control loop 1
	LAUF#1	PB2	configured control loop 1
	HALT#1	PB3	final processing after constant ope tion of control loop 1

Status markers

In addition to the call markers, the runtime system provides status markers for the runtime system itself and for every control loop in the ZVT. The status markers of the control loops (STAT#xx) are important both for the synchronization of the control blocks used there and for its initial performance.

The runtime system recognizes the following states:
- ”Reset” (STAT = -1)
- ”Initial” (STAT = 0)
- ”Running” (STAT = 1)
- ”Halt” (STAT = -2)

The control blocks do not evaluate the status of ”Reset” and ”Halt” since they are inactive. The status messages ”Initial” and ”Running” are processed by the control blocks. In the control loop status ”Initial”, the internal values are generally standardized for all the control blocks of the particular control loop and the necessary previous values for the processing are obtained so that the control block can generally continue with the existing process state smoothly and bumplessly. If the control loop status is ”Running”, all the control blocks of the particular control loop are processed according to their algorithm and operating mode.

Caution **All the control blocks should be called ”unconditionally” within their control loop (PBxx) so that they always can pass on their previous information within the process. All the control blocks can be halted using markers, so that they no longer alter their outputs. This ensures that previous information of the control blocks about the process is valid and the blocks can thus continue with the current values smoothly and bumplessly.**
Exception: The control loop is called unconditionally and the SFBs conditionally with LAUF#xx for two control-SFBs with different sampling times in one control loop.

A window containing the data of the time management table appears when the ”Closed-Loop Control” function is selected. The ZVT contains global data referring to all the control loops and local data referring to each individual control loop.

Closed loop control - editor

G l o b a l Data

Basic samp. time: 0.10 sec   global mode : active

L o c a l Data

CL.No	Reduction	Phase	Mode	Status	Samp. time
1	2	1	active		0.20 sec
2	3	1	active		0.30 sec
3	3	2	active		0.30 sec
4	3	1	p a s s i v e		0.30 sec
5	10	1	p a s s i v e		1.00 sec
6	10	1	p a s s i v e		1.00 sec
7	10	1	p a s s i v e		1.00 sec
8	15	3	p a s s i v e		1.50 sec
9	10	1	p a s s i v e		1.00 sec

Comment : Enable control loop 1

AK=RS232   NO CONNECTION

You can call a further menu with <Ctrl>+<Return>. The parameters can be modified here.

```
Commands
Global Mode
Edit comment
Edit basic sampling time
Terminate (save)
```

The individual positions of the online runtime system are explained below

Global Data

Global Mode (alterable)

- ”Edit”, ”Closed loop control”, <Ctrl>+<Return>, ”Global Mode”,

The global mode has three possible states which can be altered with a menu or with <Ctrl>+<reference characters>:

```
Global
Activate
Passivate
Norm
```

- Active: the individual control loops are processed
- Passive: the individual control loops are not processed
- Norm: the individual control loops are normalized

Activate

- ”Edit”, ”Closed loop control”, <Ctrl>+<Return>, ”Global Mode”, ”Activate”

Activation of the individual control loops or of the whole runtime system.

Activating the control loop means changing the selected control loop xx from inactive state (passive or standardizing) to active state. The marker EF_RKxx (from the ZVT) is here set to ”one” and ER_RKxx to ”zero”. The ”conditional” PBs or FBs called with the marker LAUF#xx are called according to their time condition.

Activating the global mode of the runtime system means that the individual control loops are now processed first with regard to their local modes. The status STATCLC of the whole runtime system (RTS) changes from ”Initial” to ”Running”.

If global mode is switched active and the mode belonging to the control loop xx is switched active, the marker LAUF#xx is set periodically in this state and the PBs or FBs called conditionally with this marker are called according to the time specified in the ”sampling time” field.

Passivate

- ”Edit”, ”Closed loop control”, <Ctrl>+<Return>, ”Global Mode”, ”Passivate”

Passivation of the individual control loop or of the whole runtime system.

Passivating the control loop means changing the state of the selected control loop xx from active to inactive. The marker EF_RKxx (from the ZVT) is here set to ”zero” and ER_RKxx to ”zero”. The PBs and FBs called ”conditionally” with the marker LAUF#xx are not longer called (marker LAUF#xx is no longer set according to the time condition).

Passivating the runtime system means changing all the control loops from active to inactive state. The control loops are no longer called. The marker HALTCLC is set once. A final initialization of the whole closed-loop control is possible here. The state of all the control loops as well as the state of the whole runtime system (RTS) goes to Halt.

If the global mode is set to ”passive”, the marker LAUF#xx is no longer set, regardless of the local mode of the individual control loop. The PBs or FBs called conditionally using this marker are therefore no longer called.

Normalize

- ”Edit”, ”Closed loop control”, <Ctrl>+<Return>, ”Global Mode”, ”Norm”

Normalize the individual control loops or whole runtime system.

Normalizing the control loop means changing the state of the selected control loop xx from the current state to the state in which a normalisation is possible with the marker START#xx. The marker EF_RKxx (from the ZVT) is here set to ”one” and ER_RKxx to ”one”. The marker LAUF#xx is no longer set. The PBs or FBs (control loop) called ”conditionally” with these markers are no longer called. The status of the control loop is now ”Reset”.

Normalizing the runtime system means changing the state of all the control loops from active to inactive. The state of the control loops STAT#xx as well as the state of the whole runtime system (RTS) STATCLC goes to ”Reset”. The marker STARTCLC is set to ”one” before the next command ”Activate CLC”. An initialization before the actual start of the control loops is also possible.

If the global mode is set to ”Norm”, the marker LAUF#xx is no longer set, regardless of the local mode of the individual control loop. The PBs or FBs called ”conditionally” with this marker are no longer called.

The ”passive” and ”standardize” modes are automatically passed on to all the control loops, regardless of their local modes.

The individual local modes of the control loops are valid in ”active” mode.

Edit Basic Sampling Time

- ”Edit”, ”Closed loop control”, <Ctrl>+<Return>, ”Edit basic sampling time”

The basic sampling time t0_CLC (ZVT1.3) is the time with which the closed-loop control is called. The sampling times of all the control loops are computed from this basic sampling time in connection with the parametrized reduction for the particular control loop.

The value range of this parameter in the ZVT is from 1 to 127. This corresponds to a basic sampling time from 0.01 sec to 1.27 sec.

The two positions after the decimal point must be specified.

The basic sampling time cannot be altered in online operations.

You can set a new value with ”Edit basic sampling time” (two positions after the decimal point must be specified), but in order to alter the values you must:

- stop the closed-loop control: the parameter CRT in the O-REG block is set to ”zero” (e.g. with ”Online”, ”Control List”)
- define a new value for the basic sampling time (e.g. with ”Online”, ”Closed-loop Control”, <Ctrl>+<Return>, ”Edit basic sampling time” or with online exchange in the data structure ZVT)
- activate the closed-loop control: the parameter CRT in the O-REG block is set to ”one”

Local Data

In addition to the global data, there is data for 64 control loops.

The local data are altered by selecting the column with the cursor keys. An alteration is then possible with <Return> (cursor is shifted left). The alteration is accepted with a further <Return> and it is stored with ”Terminate”.

CL.No. (alterable)

CL.No. is the number of the control loop to be displayed. During AKF configuration, any order may be defined. However, the order is sorted in increasing order after each new AKF-call.

Reduction (alterable)

The reduction is the factor with which the basic sampling time t0_CLC in the ZVT must be multiplied in order to obtain the actual sampling time of the control loop.

The effective sampling time should be selected to correspond to approx. 1/10 of the smallest process time constant to be measured.

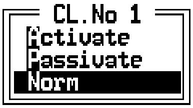
Phase (alterable)

With the phase angle you can ensure that control loops with the same reduction are called out-of-phase and not in the same call of the OBi (OB in which the closed-loop control is configured). An optimal computer workload can be attained by suitably selecting the phase angle.

In order to obtain a defined phase angle of the control loops to each other, all the control loops must be started together. The start is made with the marker CRT in the standard function block O_REG or with the global mode of the runtime system. The phase angle cannot be defined with regard to other control loops by activating a single control loop at a later time.

Mode (alterable)

Each control loop has its own mode whose contents correspond to the global mode.



Status (can only be read)

Each control loop has a status. All the control blocks belonging to the particular control loop should be connected with this status. The status informs the control blocks whether they are in initial or in running state (previous values must be initialized in initial state). The status contains four possible states:

❒ Reset
The Reset state means that an initialization of the control loop xx must be made with the call marker START#xx before the next start.

❒ Initial
The call marker LAUFxx is set once with the Initial state.With this state the previous values of the control loops are first initialized before closed-loop control starts.

❒ Running
With this state, the control loops are called periodically by the call marker LAUFxx.

❒ Halt
The Halt state means that a final initialization of the control loop xx was made with the call marker HALT#xx (if configured).

If the value specified in the Mode field does not agree with the corresponding actual status of the ZVT, the field is underlaid to show this.

Sampling Time (can only be read)

The sampling time is computed from the basic sampling time and the reduction of the particular control loop.

Edit Comment

- ”Edit”, ”Closed loop control”, <Ctrl>+<Return>,
  ”Edit Comment”

You can assign up to 40 characters of comment to each control loop with this function. The comment is automatically connected to the enable marker of the control loop EF_RKxx.

Select the required control loop with the cursor keys. The menu is then opened with <Ctrl>+<Return> or the comment is edited directly with <Ctrl>+<E>.

Terminate

- ”Edit”, ”Closed-loop Control”, <Ctrl>+<Return>, ”Terminate”

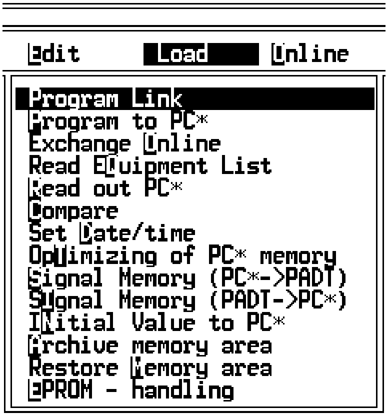
After a plausibility check, all the settings are accepted and stored.

4.4 Load

The load functions are used to communicate with the PLC and to process the user program in the PLC.

☞ **Note** Please note that all the functions except for ”Program Link” are only possible with a properly connected PLC. Information can be found in the chapter Hardware Requirements and the relevant PLC user manual.

The following load functions are provided under this pulldown menu:



4.4.1 Program Link

”Load”, ”Program Link”

The program is generated for the PLC when this function is selected.

The OB, all the blocks called by the OB, and the initial values are linked to form a program which is executable in the PLC.

When a basic software version ≥ 6.0 is used, the basic software is also linked if you have not yet done this under ”Setup”, ”Station”, ”First Use of PC*” ”BSW-Configuration”.

This program is stored in a file depending on the ”link mode” set under ”Setup”, ”Station”. It is transferred to the PLC with the function ”Program to PC*”.

4.4.2 Program to PLC

- ”Load”, ”Program to PC*”

The information to be transferred to the PLC with this function depends on the preset values.

- Basic software 6.0
  - ”Program to PC*” in the RAM
    user program, loadable basic software, equipment list and initial values are transferred
  - ”Program to PC*” in write-protected RAM (see also chapter 4.4.14)
    user programm, loadable basic software, equipment list and initial values are transferred after confirmation of the message with ”yes”
  - ”Program to PC*” in EPROM (see also chapter 4.4.14)
    equipment list and initial values are transferred after confirmation of the message with ”no”

- Basic Software < 6.0
  - ”Program to PC*” in the RAM
    User program, equipment list and initial values are transferred
  - ”Program to PC*” in write-protected RAM (see also chapter 4.4.14)
    user program, equipment list and initial values are transferred after confirmation of the message with ”yes”
  - ”Program to PC*” in EPROM (see also chapter 4.4.14)
    equipment list and initial values are transferred after confirmation of the message with ”no”

If basic software version $\geq$ 6.0 is used, the basic software is also transferred to the PLC if this was not yet done under ”Setup”, ”Station”, ”First Use of PC*” ”BSW-Configuration”.

This procedure can only be executed if the PLC is idle. The OB is loaded into those segments of the PLC which are entered under ”Setup”, ”Station” .

4.4.3 Exchange Online

- ”Load”, ”Exchange Online”
- ”Edit”, ”Block”, Block, <Return>, ”Exchange Online”

OB, PBs, FBs or data structures are replaced or extended in the PLC with this function. This is only possible if the program is running in the PLC.

If you are reaching the memory limit, an exchange might no longer be possible.

Countermeasures:
- Function ”Load”, ”Optimizing of PC* Memory” or
- change the link mode under ”Setup”, ”Station” and transfer the whole program to the PLC after relinking or
- increase the number of AKF-RAM segments under ”Setup” and transfer after a new first-time parameter assignment and linkage.

Display all existing blocks with a space and <Return> after selecting the function ”Exchange Online”. Select the block to be exchanged from these. Input is also possible with the line editor.

Several blocks can be exchanged with the input ”PB” or ”FB”. In this case, all PBs or all FBs are transmitted to the PLC.

Expert Exchange Online of FBs after Change in Declaration Part

Changes in FB declaration parts should only be made offline, i.e. with a new link/ load of user program.

Only during linkage of the AKF35 user program there is a check, whether the type and number of parameters in the declaration part of the called FBs and at the calling location (OB,PB,FB) agree. There is no check during online exchange of FBs.

However, if an online exchange cannot be avoided, you should note:

Table 15

Exchange Online of FBs after Change in Declaration Part

Required change in FB-declaration part	Exchange Online possible	Remarks to be observed
FB name or parameter name	yes	–
Insertion of parameters	yes	Insertion of ...
Setpoint type/attribute of existing parameters	cond.	Change of ...
Order of the parameters	cond.	Change of ...
Deletion of parameters	cond.	Change of ...

Insertion of Parameters

For insertion of parameters in the declaration part, please proceed as the following:

Step 1 Insert the additional parameters **at the end** of the FB declaration part.

Step 2 Exchange the FB online.

Note The new parameters may not be used initially in the FB!

Step 3 All the FB calling locations must be reparametrized corresponding to the addition.
You can determine the locations in the user program at which there are the respective FB calls using the function ”Edit”, ”Overview” or the cross-reference list.

Step 4 Exchange all reparametrized FB calls online.

Step 5 The additional instructions with which the new parameters are processed may now be inserted within the FB.

Step 6 Exchange the FB online again.

Your user program is now running with the extended FB.

Change of Setpoint Type/Attribute or Order of Existing Parameters or Deletion of Parameters
These changes can only be executed online using the following auxiliary steps:

Step 1 Copy the existing FB to another new FB using the function ”Special”, ”Copy Files”.

Step 2 Now you change the new FB as required.

Step 3 Delete the old FB call and replace it by the new FB call at all the calling locations.

Step 4 Now exchange the corresponding block online. The new FB is automatically transmitted to the PLC with the first online exchange.

Note The old FB still occupies memory space in the PLC even if it is no longer called. In order to minimize this memory space, you can delete all the networks of this FB and then exchange online.

4.4.4 Read Equipment List

- ”Load”, ”Read Equipment List”

The equipment list which was previously created and transmitted to the PLC is read into the current station and saved with this function.

An equipment list on the PADT which already exists under the current station is overwritten with this function.

Caution **Logical slot references already assigned in the PADT are lost when reading out of the PLC. (Column ”Number” in the equipment list editor). To avoid this, read out the equipment list with the function ”Read Out PC*”. (Specify: BESLIS for block list).**

4.4.5 Read Out PLC

- ”Load”, ”Read Out PC*”

This function is only possible if the link mode ”Full Representation” or ”Without Comments” was set during the last load.

Note The symbols from the SYM/COM block are not transferred and therefore cannot be read out. All IL line comments, network titles and network comments are missing if there was a previous load with the mode ”Without Comments”.

You can read out individual blocks or the equipment list from the PLC with this function, i.e. ”retranslate”. Furthermore, the equipment list and the initial values are generally read out of the PLC.

Select the required block list using the following functions:

```
                         Read PC*
Start Reading
Block List:    *
Load Block List
Store Block List
```

Start Read Out

- ”Load”, ”Read Out PC*”, ”Start Reading”

Reading out of the PLC is started with the <Return> key after specifying the blocks to be read out (poss. existing, i.e. block to be loaded).

Note The equpment list is read out by specifying ”BESLIS” for the block list.

4.4.6 Compare

- ”Load”, ”Compare”

These functions compare data of the AKF station in the PADT with those in the PLC.

Note Please note that all the functions are only possible with a properly connected PLC. Information can be found in the chapter Hardware Requirements and the relevant PLC user manual.

Note A new time is entered in the program but not in the PLC with the function ”Edit”, ”Block”, ”Exchange Online”. If the programs in the PADT and the PLC are then compared, an error message can occur although the two programs are the same. Please check both programs in this case (e.g. with read out PLC).

Blocks with PLC

- ”Load”, ”Compare”, ”Blocks with PC*”

The following functions are provided:

```
Compare Blocks
Start Comparison
Comparison Mode      Byte comparison
Title Block          Off
Title Block File     TITLE.DOC
Output filename:     A:\SYMCOM35.DOC
Block List:          *
Load Block List
Store Block List
```

Output can be to the printer, screen or to a file.

Start Comparison

- "Load", "Compare", "Blocks with PC*", "Start Comparison"
- "Load", "Compare", "Program with PC*", "Start Comparison"

The blocks of the PADT and PLC named in the block list are compared in the set mode.

The following example shows a block comparison.

```
                                  Print
Block    PADT                 PC*                  Segm  Offs  Length(BYTE)
BESLIS1  15-06-93 23:01:42    15-06-93 23:01:42    7 1000H     272 equal
OB1      16-06-93 20:15:19    16-06-93 20:15:19    7 1130H     288 equal
OB4      17-06-93 16:50:54    17-06-93 16:50:54    7 1250H     112 equal
ZVT1     18-06-93 15:45:44    18-06-93 15:41:41    7 12C0H     832 equal
FB1      16-06-93 00:08:31    16-06-93 00:08:31    7 1F20H     528 equal
IB1      18-06-93 15:41:48    18-06-93 15:41:48    7 2130H     128 equal

         Comparison OK: All blocks in PADT equal to blocks in PC
         Press any key !

<S> = print STOP,              <ESC>=Abort
```

Compare Mode

- "Load", "Compare", "Blocks with PC*", "Comparison Mode"

You can select one of the following by toggling:
Byte comparison: The blocks are compared bytewise (highest security level)
Fast comparison: Only the date and time are compared

Note If the byte comparison is "ok" and the fast comparison shows differences, the program was stored in the PLC again after the transfer.

Program with PLC

- "Load", "Compare", "Program with PC*"

Only the version date of the whole program is compared on the PADT and PLC. However, in order to ensure that all the data and program modules are identical, the function "Compare Blocks with PLC" should also be selected.

The following menu is opened when comparing "Program with PLC":

```
═══════════════ Program comparison ═══════════════
 Start Comparison
 Title Block           Off
 Title Block File      TITLE.DOC
 Output Unit           Monitor
```

The following diagram shows an example of a printout for a program comparison

```
PC*
=================
Version code      : AKF6.01
Plant/Station     : C:\AKF35E\EXERCISE
ALU-Type          : ALU 021
Generation date   : Fri Jun 18 15:41:38 1993
Date of change    : Fri Jun 18 15:41:38 1993
Max. Blocks       : 250
EPROM/RAM-Version: RAM

PADT
=================
Plant/Station     : C:\AKF35E\EXERCISE
Generation date   : Fri Jun 18 15:41:38 1993

=================
Station equal to program in PC*
```

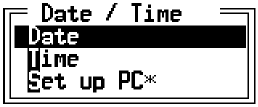
4.4.7 Set Date/Time

- ”Load”, ”Set Date/Time”

The date and time for the PADT and PLC are set in this menu.

When this function is selected, the date and time are transferred to the PLC. The marker words MW60 to MW65 are thus assigned current values.

The following functions are provided:



Set Date

- ”Load”, ”Set Date/Time”, ”Date”

You see the current date set in your PADT in the first line.
This date can correspond to today’s date if you set the date and time at system start or if your PADT has a built-in multifunction module. In this case leave the editor with the <Esc> key.
The date can be reentered in the second line.
The input must be repeated until a valid date is entered or until you leave the editor.

☞ **Note** The date of your PADT is overwritten with the entered date.
Examples of valid input: (Day,Month,Year)
12.3.1991 12 3 1991 12/3/1991 12,3,1991 12-3-1991 12:3:1991

Set Time

- ”Load”, ”Set Date/Time”, ”Time”

You see the current time set in your PADT in the first line.
This time can correspond to the current time if you set the date and time at system start-up or if your PADT has a built-in multifunction module. In this case leave the editor with the <Esc> key.
The time can be reentered in the second line. The input must be repeated until a valid time is entered or until you leave the editor with <Esc>.

Note The time of your PADT is overwritten with the entered time.
Examples of a valid input: (Hour,Minute,Second)
12.13.19 12 13 19 12/13/19 12,13,19 12-13-19 12:13:19

Set up PLC

- ”Load”, ”Set Date/Time”, ”Set up PC*”

If you just reentered the time and date, these are transferred to the PLC.
If you did not do this, the date and time of the PADT are transferred to the PLC.
MW60 to MW65 are then assigned current values in the PLC.

The date and time of the system can also be set on the PADT with the operating system functions ’Date’ and ’Time’ after switching on (or warm restart).

4.4.8 Optimizing of PLC Memory

- ”Load”, ”Optimizing of PC* Memory”

The memory organization is optimized with this function. Memory areas can become available after several ”Online Exchanges”. These unused areas are combined during optimization.

Note Scan time delays can occur during optimization.

4.4.9 Signal Memory (PLC → PADT)

- ”Load”, ”Signal Memory (PC*–>PADT)”

This function stores the signal memory contents of the PLC in a file on the PADT.

PLC programs can thus be started again with special conditions (e.g. in order to be able to start again after a change of shift from certain machine settings).

```
Read Signal Memory
Start Function
Signal list: *
File Name  : STATION  .SSP
```

The signals in the signal list which were used in the user program are taken into consideration.

Note If the function is executed for ”PC* in Scan”, only up to 18 signal states per scan are consistent.

Signal List

-”Load”, ”Signal Memory (PC*–>PADT)”, ”Signal List”
- ”Print”, ”Cross-reference List”, ”Signal List”
- ”Print”, ”Signal Occupancy List”, ”Signal List”

You must enter the signal areas for which you want to generate lists here.

The following entries are provided (with the line editor):

”*”:	all the inputs, outputs markers, marker words,... existing under the staton
”Mx-y”/”M”:	Areas or all existing markers.
”I/Qx-y”/”I/Q” :	Areas or all existing inputs/outputs.

Only the blocks of the block list are taken into consideration in the ”Print” functions

The entries ”x” and ”y” are irrelevant of which addressing, ”DIN” or ”AEG”, you selected.
The entries in this menu option must be separated by commas.

4.4.10 Signal Memory (PADT → PLC)

- ”Load”, ”Signal Memory (PADT→PC*)”

This function loads the signal memory contents already stored on the PADT back into the PLC. Only those signals which are used in the user program are taken into consideration.

PLC programs can thus be started again with special conditions (e.g. in order to be able to start again after a change of shift from certain machine settings). This is only possible if the PLC is idle.

```
═══════════════ Load Signal Memory ═══════════════
 Start Function
 File Name  : STATION  .SSP
```

4.4.11 Initial Value to PLC

- "Load", "Initial Value to PC*"

The SYM/COM block is searched for initial values. Existing initial values are loaded into the PLC.

The initial values are written into the signal and/or data structure memory of the PLC regardless of the user program. You can decide whether all the signal initial values and/or the data structure initial values used in the program are to be transferred.

This function is only possible if the PLC is idle.

```
═══════════════════ Init Value to PC* ═══════════════════
 Start Function
 Unitialisation of   :   Signalmemory (SSP)
 Data Structure List:    *
```

Initial Values to PLC

- "Load", "Initial Values to PC*", "Start Function"

After defining the other menu lines, you start the transfer to the PLC with "Start Function".

Initialisation

- ”Load”, ”Initial Values to PC*”, ”Initialisation of”

You can select the values to be initialized here. You can toggle between

- **Signal memory (SSP)**
  Only signal memory is initialized

- **Data structures**
  All the data structures specified in the data structure list are initialized

- **Signal memory + Data structures**
  Signal memory and the data structures specified in the data structure list are initialized

Data Structure List

- ”Load”, ”Initial Values to PC*”, ”Data Structure List”

You can specify a list of data structures whose initial values are to be transferred here.

The individual data structures are separated by a comma.

If * is specified, all the data structures are initialized.

4.4.12 Archive Memory Area

- ”Load”, ”Archive Memory Area”

This function creates backup copies of memory areas. An online backup can therefore be created at any time. This is particularly useful when searching for an error and when restoring after a system failure.

You can archive one or more of a total of 32 possible PLC memory areas with this function. A file under the name ’SBnn.SPB’ is set up per memory area in the defined station. Archived memory areas can be loaded into the PLC again with the function ”Restore Memory Area”.

The following functions are provided under this pulldown menu:

```
Archive memory
Start Function
From memory area:
```

Start Archiving Memory Area

- ”Load”, ”Archive Memory Area”, ”Start Function”

The specified memory areas are read out of the PLC and stored in the defined station as a file (name: SBnn.SPB). One file is set up per memory area. You can enter a comment of up to 40 characters which is also written to the file before storing the data. Files of the same memory area which are already archived are overwritten without a message.

Archive from Memory Area

- ”Load”, ”Archive Memory Area”, ”From Memory Area”

The memory areas to be archived are specified here. One or more areas may be specified. A list of all the memory areas existing in the PLC is displayed after entering a blank and <Return>. You can select a memory area from this list. If additional areas are to be selected, the list must be extended manually. All the existing memory areas are archived if ”*” is entered.

Examples:	29	SB 29
	16,18-20	SB 16,18,19,20
	*	all existing SBs in the PLC

4.4.13 Restore Memory Area

- ”Load”, ”Restore Memory Area”

You can load memory areas which were already archived back into the PLC with this function. One or more memory areas can be restored with one call. Select the source and target memory areas with the following functions and start the function.

```
Restore memory
Start Function
From memory area:
To memory area:
```

Start Restoring Memory Area

- ”Load”, ”Restore Memory Area”, ”Start Function”

The specified memory areas are loaded into the PLC after an inquiry. The number of source and target areas must agree. Furthermore, the target areas must be at least as large as the source areas. You can abort with <Esc> between two copy procedures. If the PLC is in the scan, AKF segments and segments in which a Dolog B Vlist is running cannot be restored. The PLC must be halted in this case.

Restore from Memory Area

- ”Load”, ”Restore Memory Area”, ”From Memory Area”

The source memory areas to be restored are entered here. One or more areas may be entered. A list of all the archived memory areas is displayed with the relevant comment and corresponding length specification after entering a blank and <Return>. An element can be selected from this list. If additional memory areas are selected, the list of the source memory areas must be extended manually. All the existing memory areas can be restored by entering ”*”.

Examples:	29	from SB 29
	16,18-20	from SB 16,18,19
	*	all archived memory areas

Restore to memory area

- ”Load”, ”Restore Memory Area”, ”To Memory Area”

The target memory areas are entered here. Their length must be greater than or equal to the source memory areas. A list of all the memory areas existing in the PLC is displayed by entering a blank and <Return>. An element can be selected from the list. If several memory areas are to be restored, the list of the target memory areas must be extended manually. The archived memory areas are written back to the original memory areas by entering ”*”, i.e. the archivation file ’SB01.SPB’ is written back into memory area 1 of the PLC etc.

If the PLC is in the scan, AKF segments and segments in which a Dolog B Vlist is running cannot be restored. These memory areas no longer appear in the selection list.

Examples:	29	To SB 29
	16,18-20	To SB 16,18,19,20
	*	Into original memory areas

4.4.14 EPROM handling

- ”Load”, ”EPROM-handling”

Preparations for EPROM creation are made in AKF35. The programming of the EPROMs is carried out with the EPROM EPS 2000 or EPS 386 programming panels and their special user software.

The following data can be transferred to EPROM:

BSW < 6.0	File	CODE.SPS (created during linkage)
	Files	SBnnn.SPB, without the first 64 bytes (archive memory area)
BSW $\geq$ 6.0	File	CODE.SPS (created during linkage)
	File	GSW.SPS (created during linkage)
	Files	SBnnn.SPB, without the first 64 bytes (archive memory areaa)

Expert The last diskette of the AKF35 software contains the file ”SPLIT.EXE”. With this program you can split large files (e.g. CODE.SPS) into files of any size. The EPROM size can thus be specified (e.g. 128 kbyte for ALU 071, 256 kbyte for ALU 021). This program is started with the command ”`A:\SPLIT`”. Enter the required definitions and follow the text on the screen.

The following chapter is divided into steps which are carried out in AKF35 and in the EPROM programming software.

The user program is generally first created and loaded as a RAM version, also for an EPROM version.

It is not possible to program EPROMs directly with AKF35; you must use the program ”EPS2000” (ALU 821, ALU 150, ALU 286, ALU 011, ALU 021) or ”EPS386” (ALU 061, ALU 071).

Note Please also note the descriptions of the programming stations

EPS 2000 (SET) E no. 424-240376
EPS 386 E no. 424-271571

❒ Work Steps with AKF35/EPS xx to Operate the User Program in EPROM

Step 1 Poss. prepare hardware for EPROM operation (e.g. jumper E5 for ALU 150)

Step 2 Switch to EPROM version in the menu ”Setup”, ”First Use of PC*” and carry out the first-time parameter assignment again after defining the EPROM and the RAM segment(s)

☞ **Note** At least one segment must be entered for ”AKF-RAM Segment Nos.” (status display, data structures with attribute ”write” etc.)

Step 3 Link program (the files of the user program, CODE.SPS and poss. of the loadable basic software, GSW.SPS, are created)

Step 4 Leave the AKF35 program (”Special”, ”End of Station Handling”)

Step 5 Connect EPS to COM1 interface of PADT and start EPROM program. (Call: ”EPS2000” or ”EPS386”)

Program the file CODE.SPS and poss. GSW.SPS (loadable BSW) on EPROM

Step 6 Select EPROM type (only for EPS2000),

Step 7 Set 16-bit mode (only for EPS2000)

Step 8 Call ’Disk → EPROM’ function

Step 9 Enter file name
AKF user program: File ”CODE.SPS” or
loadable basic software (as of version 6.0, for ALU 021 and ALU 071): File ”GSW.SPS”

Step 10 Start EPROM (low byte) as of 0

The EPROM is programmed. After the request:

Step 11 Insert 2nd EPROM (high byte)

Step 12 Call ’Verify’ function

The file contents are compared with the EPROM contents.

Step 13 Insert EPROMs on the segments specified in the AKF program

Step 14 Remove PADT ↔ EPS connecting cable and insert PADT ↔ PLC

Step 15 Call AKF35 program again

Step 16 Load equipment list and initial values to the PLC with the ”Program to PC*” function (answer inquiry with ”N” for no).

Step 17 Start program in PLC

The AKF programming is now terminated and you can remove the PADT ↔ PLC connecting cable.

❐ Work Steps in AKF35 to Operate the User Program in Write-Protected RAM

Step 1 Set write protection to hardware (e.g. DIP-switch)

Step 2 Switch to EPROM version in the menu ”Setup”, ”First Use of PC*” and carry out the first-time parameter assignment again after defining the EPROM and RAM segment(s)

☞ **Note** The RAM segments which are write-protected must be entered at ”AKF-EPROM Segment Nos.”.

Step 3 Link program (the files of the user program, CODE.SPS and poss. the loadable basic software, GSW.SPS, are created)

Step 4 Remove write protection

Step 5 Carry out ”Program to PC*” function, answer inquiry about write-protected RAM with ”y” for yes

Step 6 Set write protection to hardware

Step 7 Start program in PLC

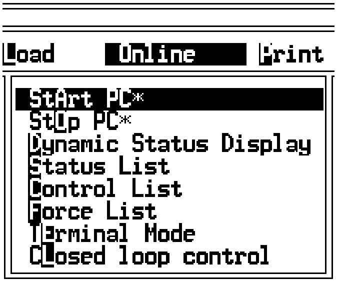
The AKF programming is now terminated and you can remove the PADT ↔ PLC connecting cable.

4.5 Online

The online functions are used to test the program in the PLC.

Note Please note that all the functions are only possible if the PLC is properly connected. Information can be found in the chapter on hardware requirements and in the relevant PLC user manual.

The following functions can be selected:



4.5.1 StArt PLC

- ”’Online”, ”StArt PC*”

The program in the PLC is started with this function. ”PC* ACTIVE” appears at the lower right edge of the screen.

The initial start is also made here after ”First Use of PC*” with the Dolog AKF software.

If the initial parametrization was made in terminal mode, the initial start must also be made in terminal mode with the Bsdol function ”START”.

4.5.2 StOp PLC

- ”Online”, ”StOp PC*”

The program in the PLC is stopped with this function. Note that intermediate results might be lost.

4.5.3 Dynamic Status Display

- ”Online”, ”Dynamic Status Display”

You can display the dynamics of a program in the PLC with this function.

The following functions are displayed when this menu line is selected:

```
Dyn. Status Display
Online Recording
Single Shot Processing
Triggered Recording
Output Mode        FBD
Address Mode       DIN
Addressing         SYM
```

Online Recording

- ”Online”, ”Dynamic Status Display”, ”Online Recording”

This display works with running actualization of signal changes.

You can select the block to be displayed from the window after entering a space and <Return>. You can also specify a block name directly with the line editor instead of entering spaces. Start the running display by pressing <Return>.

Caution This status display is only possible for PLC active!

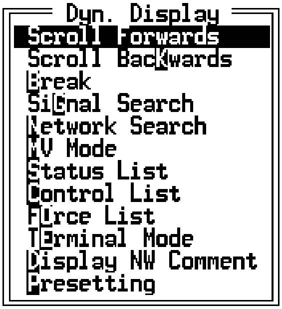
Dynamic Status Display

- ”Online”, ”Dynamic Status Display”, ”Online Recording”, Block, <Return>
- ”Edit”, ”Block”, Block, <Return>, ”Dyn. Status Display”, <Return>
- ”Edit”, ”Overview”, Select Block, ”Edit Block”, <Return>, ”Dyn. Status Display”, <Return>

A pulldown menu now appears with which you can execute different operations during the dynamic display.

You are in dynamic display mode.

a) You can select the following functions in the pulldown menu after pressing the <Return> key:



b) You can always call the functions outside the pulldown menu with <Ctrl>+<reference character>, e.g. ”Si**G**nal Search” with <Ctrl>+<G>.
You can scroll forwards network by network with <PgDn>, scroll backwards with <PgUp> and abort with <Esc>. You can scroll by half a screen page in the network with <Ctrl>+<PgDn> or <Ctrl>+<PgUp>.

Break Dynamic Status Display

- ”Online”, ”Dynamic Status Display”, ”Online Recording”, Block, <Return>, ”Break”
- ”Edit”, ”Block”, Block, <Return>, ”Dynamic Status Display”, <Return>, ”Break”
- ”Edit”, ”Overview”, Select Block, ”Edit Block”, <Return>..

The dynamic display is ended when this function is selected.

You can activate this function in the window by pressing the <Return> key.

You close the window again with the <Esc> key.

Dynamic Status Display MV-Mode

- ”Online”, ”Dynamic Status Display”,
  ”Online Recording”, Block, <Return>, ”MV Mode”
  or ”Single Shot Processing”, ”Start Single Shot”, <Return>,...
  or ”Triggered Recording”, ”Start Display”, <Return>,...
- ”Edit”, ”Block”, Block, <Return>, ”Dyn. Status Display”, <Return>,...
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”, <Return>...

IL
The symbol and the comments of the signal selected with the cursor are always displayed in the status line.
The following are displayed in the IL line:

MV-Mode = off	the signal status in decimal and the line comment
MV-Mode = on	the signal status in decimal and binary

LD/FBD
The following are displayed in the status line:

MV-Mode = off	symbol and comment of the selected signal
MV-Mode = on	signal and signal status in decimal, binary, hexadecimal and ASCII

You close the window again with the <Esc> key.

Status List, Control List, Force List

You can process the lists within the dynamic status display with these functions. They are described under the individual functions in the ”Online” menu.

Terminal Mode

You switch over to terminal operation (online) with this function. The function is described under chapter 4.5.7 on page 255 in the ”Online” menu.

Display NW Comment

- ”Online”, ”Dynamic Status Display”,
  ”Online Recording”, Block, <Return>, ”Display NW Comment”
  or ”Single Shot Processing”, ”Start Single Shot”, <Return>,...
  or ”Triggered Recording”, ”Start Display”, <Return>,...
- ”Edit”, ”Block”, Block, <Return>, ”Dyn. Status Display”, <Return>,...
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”, <Return>...

The network comment created in a window under ”Edit” can be called at this location.

Dynamic Status Display Presetting

- ”Online”, ”Dynamic Status Display”,
  ”Online Recording”, Block, <Return>, ”Presetting”
  or ”Single Shot Processing”, ”Start Single Shot”, <Return>,...
  or ”Triggered Recording”, ”Start Display”, <Return>,...
- ”Edit”, ”Block”, Block, <Return>, ”Dyn. Status Display”, <Return>,...
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”, <Return>...

You can select the following functions at this location:

IL/LD/FBD
ABS/SYM

Single Shot Processing

- ”Online”, ”Dynamic Status Display”, ”Single Shot Processing”

The statuses of the entered signals are recorded for one scan.

You can select the following functions in this pulldown menu:

```
Single Shot
StArt Single Shot
Block List:     *
Status List
Control List
Force List
```

Start Single Shot

- ”Online”, ”Dynamic Status Display”, ”Single Shot Processing”, ”Start Single Shot”

You first define the signals to be recorded using the block and status list. You can also set certain signals to defined values using the control and force lists.

Caution The PLC must be in Stop for this function.

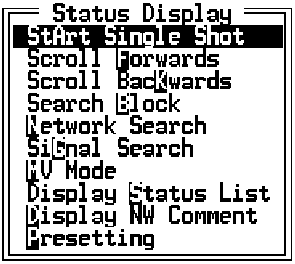
The first recording is then started with this function. The first network of the required block then appears on the screen.

Caution The PLC executes the first 5 norming scans without peripheral operations. The actualization of the I/Os and of the networking operations is stopped during this function. All the ”background functions” are affected (e.g. remote operation)

Single Shot Status Display

”Online”, ”Dynamic Status Display”, ”Single Shot Processing”, ”Start Single Shot”, <Return>

a) You can now select the following (menu at the right edge of the screen):



b) You can always call the functions outside the pulldown menu with <Ctrl>+<reference character>, e.g. ”Search **B**lock” with <Ctrl>+<B>.
You can scroll forwards network by network with <PgDn> and backwards with <PgUp>. You can scroll by half a screen page in the network with <Ctrl>+<PgDn> or <Ctrl>+<PgUp>.

The single shot display is ended with <Esc>.

The number of recordings is faded into the right side of the lowest line together with the block and network numbers.

Start Single Shot

- ”Online”, ”Dynamic Status Display”, ”Single Shot Processing”, ”Start Single Shot”, <Return>, ”Start Single Shot”

You can start the PLC for a further instruction list scan with this function. The recording number at the lower edge of the screen is incremented.

You cannot look at the signal statuses of older recordings. (Status test as for Online Recording.)

Search Block

- ”Online”, ”Dynamic Status Display”, ”Single Shot Processing”, ”Start Single Shot”, <Return>, ”Search Block”
- ”Online”, ”Dynamic Status Display”, ”Triggered Recording”, ”Start Display”, <Return>, ”Search Block”

A window appears containing all the blocks in the block list. The next block required can be selected with the cursor keys and <Return>, so that you can look at its status in the same scan.

Display Status List

- ”Online”, ”Dynamic Status Display”, ”Single Shot Processing”, ”Start Single Shot”, <Return>, ”Display Status List”

A table containing the actual values of the signals in decimal, binary, ASCII and hexadecimal (depending on the signal type) is displayed when this function is selected.

Status List, Control List, Force List

You can pocess the lists within the dynamic status display with these functions. The descriptions are contained under the individual functions in the ”Online” menu.

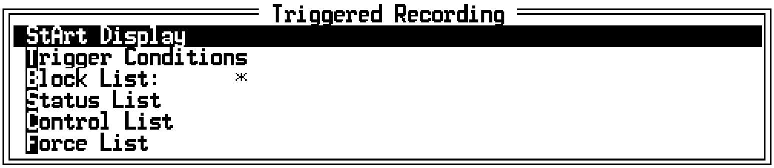
Triggered Recording

- ”Online”, ”Dynamic Status Display”, ”Triggered Recording”

The signal statuses of selected signals are recorded for several scans.

Caution The PLC must be active for this function.

The following functions can be executed in the pulldown menu:



Start Display

- ”Online”, ”Dynamic Status Display”, ”Triggered Recording”, ”Start Display”

The function is started with this menu line when the signals to be recorded and the trigger conditions have been defined.

Two windows first appear. You can abort the function during recording with the first window (with the <Esc> key). The second window shows the ready signal with the number of recordings made.
Network 1 of the first block (first recording) is then faded in.

Triggered Recording

- ”Online”, ”Dynamic Status Display”, ”Triggered Recording”, ”Start Display”, <Return>

a) A window with the following functions appears at the right edge of the screen:

b) You can always call the functions outside the pulldown menu with <Ctrl>+<reference character>, e.g. ”Search **B**lock” with <Ctrl>+<B>.

The triggered recording is ended with <Esc>.

Next Recording

- ”Online”, ”Dynamic Status Display”, ”Triggered Recording”, ”Start Display”, <Return>, ”Next Recording”

You can select the next recordings with this function. The recording number is at the lower right of the lowest line of the screen.

Previous Recording

- ”Online”, ”Dynamic Status Display”, ”Triggered Recording”, ”Start Display”, <Return>, ”Previous Recording”

You can select previous recordings with this function. The recording number is at the lower right of the lowest line of the screen.

Select Recording

- ”Online”, ”Dynamic Status Display”, ”Triggered Recording”, ”Start Display”, <Return>, ”Select Recording”

You can display the recording of your choice with this function. The recording number is at the lower right of the lowest line of the screen.

Display Status List

- ”Online”, ”Dynamic Status Display”, ”Triggered Recording”, ”Start Display”, <Return>, ”Display Status List”

A table containing the actual values (current recording) of the signals in decimal, ASCII and hexadecimal (depending on the signal type) is displayed when this function is selected. Signal statuses from several recordings can be obtained by scrolling.

Trigger Conditions

- ”Online”, ”Dynamic Status Display”, ”Triggered Recording”, ”Trigger Conditions”

Different conditions must be defined in the ”Trigger Conditions” menu in order to execute a trigger:

- the condition under which the recordings should be made
- the condition for the start of the recording (start condition)
- the condition for the stop of the recording (stop condition).

The following pulldown menu is provided:

```
                Trigger Conditions                    Conditions
1 Record. Condition :  Unconditioned           1 Unconditional
2         Signal    :  M1.1                    2 Status 0
3 Start Condition   :  Unconditioned           3 Status 1
4         Signal    :  M1.1                    4 Change 0->1
5 Stop  Condition   :  Unconditioned           5 Change 1->0
6         Signal    :  M1.1                    6 Any change
7 No. of Overflows  :  0
```

One function each is selected from the right window for the recording, start and stop conditions and stored in the left window with <Return>. The relevant signal address is then specified (symbolic or absolute). If the condition ”unconditional” is set, this may be omitted. Signal specifications which already exist are then ignored.

The number of overflows is also entered (0 ... N).

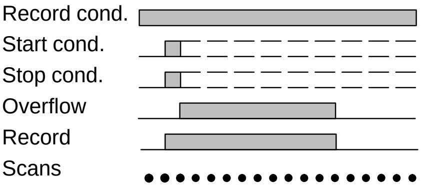
The menu is stored by pressing the <Esc> key if all the entries are correct.

Examples of Triggered Status Display

☞ **Note** In the diagrams for the timing of the different conditions, ”satisfying” the specified condition (yes or no) is displayed and **not** the value of the signal (0 or 1).

a) Triggering to ”1”-status, with overflows

Clock diagram

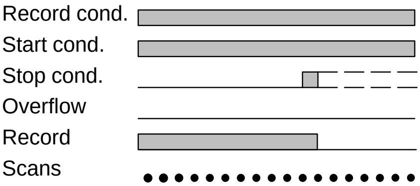


Trigger conditions

Recording condition	unconditional
Recording signal	
Start condition	Status 1
Start signal	M4.4
Stop condition	Status 1
Stop signal	M4.4
No. of overflows	10

b) Pretriggering at positive edge

Clock diagram

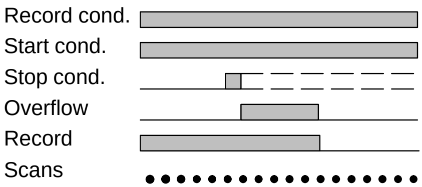


Trigger conditions

Recording condition	unconditional
Recording signal	
Start condition	unconditional
Start signal	
Stop condition	Change 0-1
Stop signal	M4.5
No. of overflows	0

c) Midtriggering at positive edge, overflows

Clock diagram

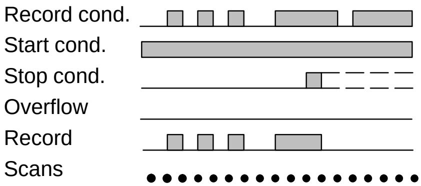


Trigger conditions

Recording condition	unconditional
Recording signal	
Start condition	unconditional
Start signal	
Stop condition	Change 0-1
Stop signal	M4.6
No. of overflows	5

d) Recording of all changes of marker M4.7, until marker M4.8 has status ”1”. If the memory is not sufficient to store all recordings, at least the events can be counted.

Clock diagram



Trigger conditions

Recording condition	Any change
Recording signal	M4.7
Start condition	unconditional
Start signal	
Stop condition	Status 1
Stop signal	M4.8
No. of overflows	0

Status List, Control List, Force List

You can process the lists within the dynamic status display with these functions. The descriptions can be found under the individual functions in the ”Online” menu.

4.5.4 Status List

– ”Online”, ”Status List”
- ”Online”, ”Dynamic Status Display”,
  ”Online Recording”, Block, <Return>, ”Status List”
  or ”Single Shot Processing”, ”Status List”
  or ”Triggered Recording”, ”Status List”
- ”Edit”, ”Block”, Block, <Return>, ”Dyn. Status Display”, <Return>,...
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”, <Return>...

The status of signals is displayed in the PLC with this function.
If possible, the signal contents are displayed here in all display modes: decimal, binary (1...32 bit), hexadecimal and ASCII.

Signal	Decimal	Binary	HEX	ASCII
Mxx.xx	-	+	-	-
MWxxxxx	+	+	+	+
MDxxxxx	+	+	+	+
MFxxxxx	+	-	-	-
I/Qxx.xx	-	+	-	-
Symbol	-	+	-	-

You can display the status of any max. 18 signals in table form.
+ means: display possible for the corresponding signal type.

The following input is possible for settings of:

	DIN	AEG
Inputs	I2.1 ... I160.32	I2A2 ... I160E32
Outputs	Q2.1 ... Q160.32	Q2A2 ... Q160E32
Marker (Bit)	M1.1 ... M313.16	M1 ... M10 000
Marker word	MW1 ... 10 000	
Marker double word	MD1 ... 9 999	
Marker floating point word	MF1 ... 9 999	

Symbol names can be entered for the corresponding address mode.

Pressing the <F1> key starts the status display. The <Esc> key ends the function.

4.5.5 Control List

– ”Online”, ”Control List”
- ”Online”, ”Dynamic Status Display”,
  ”Online Recording”, Block, <Return>, ”Control List”
  or ”Single Shot Processing”, ”Control List”
  or ”Triggered Recording”, ”Control List”
- ”Edit”, ”Block”, Block, <Return>, ”Dyn. Status Display”, <Return>,...
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”, <Return>...

Signals are set to the defined value in the PLC for one scan with this function. Input is possible in four display modes if permitted in the table below.

The window displays the status of these signals after they have been passed to the PLC.

In this way you can control the status of max. 18 signals in the PLC for one scan each. The status display follows.

Signal	Decimal	Binary	HEX	ASCII
Mxx.xx	-	+	-	-
MWxxxxx	+	+	+	+
MDxxxxx	+	+	+	+
MFxxxxx	+	-	-	-
I/Qxx.xx	-	+	-	-
Symbol	-	+	-	-

+ means: display possible for the corresponding signal type.

The following input is possible for settings of:

	DIN	AEG
Inputs	I2.1 ... I160.32	I2A2 ... I160E32
Outputs	Q2.1 ... Q160.32	Q2A2 ... Q160E32
Marker	M1.1 ... M313.16	M1 ... M10 000
Marker word decimal	MW1 ... 10 000	
Marker double word	MD1 ... 9 999	
Marker floating point word	MF1 ... 9 999	

Symbol names can be entered for the corresponding address mode.

Pressing the <F1> key starts the control display. The <Esc> key ends the function.

☞ **Note** It is recommended that you do not change the actual values of memory and counters with the control or force lists.

4.5.6 Force List

– ”Online”, ”Force List”
- ”Online”, ”Dynamic Status Display”,
  ”Online Recording”, Block, <Return>, ”Force List”
  or ”Single Shot Processing”, ”Force List”
  or ”Triggered Recording”, ”Force List”
- ”Edit”, ”Block”, Block, <Return>, ”Dyn. Status Display”, <Return>,...
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”, <Return>...

You can enter I/O signals with a fixed assigned valence in this list. You can switch forcing on and off with the function keys.

The defined input valences are passed to the program independently of the position of the contacts after switching on. The defined output valences are passed to the outputs independently of the results of the operations.

You can force up to 18 I/O signals in the PLC.

Signal	Decimal	Binary	HEX	ASCII
I/Qxx.xx	-	+	-	-
Symbol	-	+	-	-

+ means: display possible for the corresponding signal type.

The following input is possible for settings of:

	DIN	AEG
Inputs	I2.1 ... I160.32	I2A2 ... I160E32
Outputs	Q2.1 ... Q160.32	Q2A2 ... Q160E32

Symbol names can be entered for the corresponding address mode.

The <F1> key starts the force. The <F2> key switches it off.
The <Esc> key ends the function.

Note It is recommended that you do not change the actual values of memory and counters with the control or force lists.

4.5.7 Terminal Mode

– ”Online”, ”Terminal Mode”
- ”Online”, ”Dynamic Status Display”,
  ”Online Recording”, Block, <Return>, ”Terminal Mode”
  or ”Triggered Recording”, ”Start Display”, ”Terminal Mode”
- ”Edit”, ”Block”, Block, <Return>, ”Dyn. Status Display”, <Return>,...
- ”Edit”, ”Overview”, Select Block, <Return>, ”Edit Block”, <Return>...

Caution This function is only possible for a RS232 connection.

The terminal operations with the PLC are started here. The operating functions of the Bsdol operating system of the PLC are provided here.

The function is left with <F9>.

- **BSW ≥ 5.05**
  You must execute the Bsdol function ”SSN” to norm the system memory for an initial start-up of the PLC and after power failure in the ALU without battery backup. ”Terminal Mode” must be selected before ”First Use of PC*” in this case. If the terminal mode does not signal ”Dolog B”, plug in and pull out the reset pin of the ALU.

The ”SSN” function is described in:

[5)]A500
Grundsoftware Version 6.0
Benutzerhandbuch
A91M.12-279344

Note The most important functions can be found in the helptexts. A complete list of the functions can be found in the relevant PLC user manuals (Bsdol operating functions).

5) in german language

4.5.8 Closed Loop Control

- ”Online”, ”Closed loop control”

Note This function is only possible if the user program in the PLC contains the standard function block O-REG with the time management table (ZVT).

A window containing the data of the time management table appears when this function is selected. The ZVT contains global data referring to all the control loops.

```
                         Closed loop control
                         G l o b a l   Data
        Basic samp. time:  0.10 sec        global  mode   :     active
Runtime exceeded          :  0             Max RVL-runtime  :  0.00 sec

 running  CL : 6      halting  CL : 58      CL to be normed    : 0

                         L o c a l   Data
CL.No    Reduction      Phase        Mode            Status      Samp. time

  1          2            1          active          running      0.20 sec
  2          3            1          active          running      0.30 sec
  3          3            2          active          running      0.30 sec
  4          3            1       p a s s i v e      h a l t      0.30 sec
  5         10            1       p a s s i v e      h a l t      1.00 sec
  6         10            1       p a s s i v e      h a l t      1.00 sec
  7         10            1       p a s s i v e      h a l t      1.00 sec
  8         15            3       p a s s i v e      h a l t      1.50 sec
  9         10            1       p a s s i v e      h a l t      1.00 sec

 Comment   :  Enable control loop 1                    Connection : Stop
```

You can call a further menu with <Ctrl>+<Return>. You can modify the changeable parameters here.

```
        Commands
Global Mode
Edit comment
Edit basic sampling time
Terminate (save)
```

An explanation of the individual positions of the online runtime system follows

Global Data

Global Mode (changeable)

- ”Online”, ”Closed loop control”, <Ctrl>+<Return>, ”Global Mode”,
- ”Online”, ”Closed loop control”, <Ctrl>+<Return>, ”Global Mode”, ”Activate” or ”Passivate” or ”Norm”

Global mode has three possible statuses, which can be changed with a menu or with <Ctrl>+<reference character>:

- ❒ active: the individual control loops are processed
- ❒ passive: the individual control loops are not processed
- ❒ norm: the individual control loops are normed

The ”passive” and ”norm” modes are automatically passed on to all the control loops, irrelevant of their local modes.

The individual local modes of the control loops are valid in ”active” mode.

Edit Basic Sampling Time

- ”Online”, ”Closed loop control”, <Ctrl>+<Return>, ”Edit Basic Sampling Time”

The basic sampling time t0_CLC (ZVT1.3) is the time with which the closed-loop control is called. The sampling times of all the control loops are computed from this basic sampling time in connection with the reduction parametrized for each control loop.

The value range of this parameter in the ZVT is from 1 to 127. This corresponds to a basic sampling time of 0.01 sec to 1.27 sec. If 0.01 to 1.27 is entered, two positions must be specified after the decimal point.

The basic sampling time should be selected so that it is approx. 1/10 of the smallest process time constant to be measured.

You may set a new value with ”Edit Basic Sampling Time” (two positions must be specified after the decimal point), but to change the values:

- the closed-loop control must be stopped: the parameter CRT at block O-REG is set to ”zero” (e.g. with ”Online”, ”Control List”)
- a new value must be defined for the basic sampling time (with ”Online”, ”Closed loop control”, <Ctrl>+<Return>, ”Edit Basic Sampling Time”)
- the closed-loop control must be activated: the parameter CRT at block O-REG is set to ”one”

Runtime Exceeded (can only be read)

The runtime is exceeded if the current processing time of the individual control loops is greater than the parametrized basic sampling time t0. The closed-loop control cannot be called again while it is still active.

The number of times that the runtime is exceeded is summed up by the marker CRT in block O-REG (CRT changes from ”zero” to ”one”) when the activation of the closed-loop control is started and is displayed in this marker word.

Maximum RVL-Runtime (can only be read)

The maximum runtime of the control loops computed since the beginning of closed-loop control (maximum runtime of the OB in which the control loop is configured) is displayed here. The current number of control loops is also displayed with the following states:

- running control loops (CL, can only be read)
- halting control loops (CL, can only be read)
- control loops to be normed (CL, can only be read)

Local Data

The data of the 64 control loops are also displayed in addition to these global data.

The local data are modified by selecting the column with the cursor keys. A modification is then possible with <Return> (cursor is shifted left). The modification becomes effective with a further <Return> and the data are stored with ”Terminate (Save)”.

CL.No (can be changed)

The CL.No is the number of the control loop to be displayed. Any order can be defined during AKF configuration, but they are sorted in increasing order after each new AKF call.

Reduction (can be changed)

The reduction is the factor with which the basic sampling time t0 from the ZVT must be multiplied in order to obtain the actual sampling time of the control loop.

Phase (can be changed)

With the phase angle you can prevent control loops with the same reduction from being called in the same call of the O-REG (OB in which the closed-loop control is configured). They are then called out of phase. An optimal use of the computer can be attained with a suitable phase angle.

Mode (can be changed)

Each control loop has its own mode which has the same contents as the global mode.

Status (can only be read)

Each control loop has a status. All the control blocks belonging to the particular control loop should be connected with this status. The status tells the control blocks whether they are in intial or in running state (previous values must be initialized in initial state). The status contains four possible states:

- Reset

The Reset state means that an initialization of the control loop xx can be made before the next start using the call marker START#xx.

- Initial

- Running

- Halt

The Halt state means that a final initialization of the control loop xx can be made with the call marker HALT#xx (if configured).

If the value specified in the field does not agree with corresponding current status of the ZVT, the field is underlined to show this.

This state generally only occurs if

- CRT = 0
- the PLC is in Stop
- the global mode is not ”running”
- a modification was just made (short-term Halt)

Sampling Time (can only be read)

The sampling time is computed from the basic sampling time and the reduction of the relevant control loop.

Edit Comment

- ”Online”, ”Closed loop control”, <Ctrl>+<Return>, ”Edit Comment”

You can assign up to 40 characters of comment to each control loop with this function. The comment is automatically connected to the enable marker of the control loop EF_RKxx.

Select the required control loop with the cursor keys. The menu is then opened with <Ctrl>+<Return> or the comment is edited directly with <Ctrl>+<C>.

Terminate (Save)

- ”Online”, ”Closed loop control”, <Ctrl>+<Return>, ”Terminate (Save)”

All the settings are accepted and stored after a plausibility check.

4.6 Print

You can output the following lists and files on the printer, in a file or on the monitor.

```
line    Print    Special

Overview
Program Log
Symbols and Comments
Equipment List
Cross Reference List
Signal Occupancy List
Station Setup
Command File
Signal Memory Contents
Data Structure
```

4.6.1 Overview

- ”Print”, ”Overview”

You can print a survey of the existing graphic program execution here with ”Edit”, ”Overview”.

The following pulldown menu is provided after selection:

```
Print Overview
Start Print
Title Block          Off
Title Block File     TITLE.DOC
Output filename:     A:\SYMCOM35.DOC
Start Page Number:   1
```

Start Print

- ”Print”, ”Overview”, Start Print”
- ”Print”, ”Program Log”, ”Start Print”
- ”Print”, ”Symbols and Comments”, ”Start Print”
- ”Print”, ”Equipment List”, ”Start print”
- ”Print”, ”Cross-reference List”; ”Start Print”
- ”Print”, ”Signal Occupancy List”, ”Start Print”
- ”Print”, ”Setup Station”, ”Start Print”
- ”Print”, ”Command File”, ”Start Print”
- ”Print”, ”Signal Memory Contents”, ”Start Print”
- ”Print”, ”Data Structures”, ”Start Print”
- ”Setup”, ”Station”, ALU 0xx, ”First Use of PC*”, ”BSW Configuration (BSW>=V6.0)”, Module, ”Print”, ”Start Print”

You can start the output depending on the remaining entries in the pulldown menu.

Start Page Number

- ”Print”, ”Overview”, ”Start Page Number”
- ”Print”, ”Program Log”, ”Start Page Number”
- ”Print”, ”Symbols and Comments”, ”Start Page Number”
- ”Print”, ”Equipment List”, ”Start Page Number”
- ”Print”, ”Cross-reference List”, ”Start Page Number”
- ”Print”, ”Signal Occupancy List”, ”Start Page Number”
- ”Print”, ”Setup Station”, ”Start Page Number”
- ”Print”, ”Command File”, ”Start Page Number”
- ”Print”, ”Signal Memory Contents”, ”Start Page Number”
- ”Print”, ”Data Structures”, ”Start Page Number”
- ”Setup”, ”Station”, ALU 0xx, ”First Use of PC*”, ”BSW Configuration (BSW>=V6.0)”, Module, ”Print”, ”Start Page Number”

You can enter the page number (1 to 9999) for the first page of the print output with this function.

You begin again for each block or each list with blockwise page mode.
The output is numbered continuously for continuous page mode.

4.6.2 Program Log

- ”Print”, ”Program Log”

With this function, you can output your user program (OB, PB, FB) created under ”Edit”, ”Blocks” in IL, LD or FBD to the printer, file or monitor.

You can select and call the following functions in this pulldown menu:

```
                         Print Program Log
Start Print
Output Mode                  IL
Symbols and Comments         On
Local Cross Refer. List      On
Title Block                  On
Title Block File             TITLE.DOC
Output filename:             STATION.LST
Block List:                  PB
Load Block List
Store Block List
Page Mode                    blockwise
Start Page Number:           1
```

Output Mode

- ”Print”, ”Program Log”, ”Output Mode”
- ”Online”, ”Dyn. Status Display”, ”Output Mode”

You can select the mode of the output of your program log in this menu option. You can select instruction list (IL), ladder diagram (LD) or function block diagram (FBD). You can switch by toggling. (<Return>)

Symbols and Comments

- ”Print”, ”Programm Log”, ”Symbols and Comments”

You can toggle between ”on” and ”off”.

The entry ”on” means that the symbolic names and comments assigned to the individual signals (inputs, outputs, markers, ...) appear in the output list next to the signals.
The entry ”off” prevents occurrence of these symbolic names and comments in the program log.

Local Cross-reference List

- ”Print”, ”Program Log”, ”Local Cross-Reference List”

All the inputs, outputs, markers, ... which you used are listed in the local cross-reference lists (in the blocks according to the blocks list). The networks of the particular blocks in which the addresses occur are specified here.

You can toggle between ”on” and ”off”.

The entry ”on” means that the local cross-reference list appears in the program log. In the printout, the networks in which the signals occur are also printed for each block.

IL/LD/FBD	e.g. entry ”:1 I” or ”:5 O”. i.e. the required signal appears in network 1 as Input $\hat{=}$ I or in network 5 as Output $\hat{=}$ O.

Page Mode

- ”Print”, ”Program Log”, ”Page Mode”
- ”Print”, ”Cross-Reference List”, ”Page Mode”
- ”Print”, ”Signal Occupancy List”, ”Page Mode”
- ”Print”, ”Command File”, ”Page Mode””

With this function you define how the pages are to be numbered by toggling:

continuous	the start page number is only defined at the beginning, counting is continuous throughout the different blocks
blockwise	numbering is begun again for each new block / each new list (for start page number)

4.6.3 Symbols and comments

- ”Print”, ”Symbols and Comments”

You can print the signals (input, output, marker,..) to which you assigned symbolic names and comments in the menu option ”Edit”, ”Symbols and Comments”.

The following functions are provided with this pulldown menu:

```
Print Symbols and Comments
Start Print
Title Block           Off
Title Block File      TITLE.DOC
Output Unit           Monitor
Signal List:          *
Start Page Number:    1
```

4.6.4 Equipment List

- ”Print”, ”Equipment List”

The equipment list of your station is printed with this function. The slot reference, configuration, BIK number, structure types, number, directory block and comments are output here. The file is only printed up to the last allocated element.

The following functions are provided with this pulldown menu:

```
Print Equipment List
Start Print
Title Block                 Off
Title Block File            TITLE.DOC
Output Unit                 Monitor
Start Page Number:          1
```

4.6.5 Cross-Reference List

- ”Print”, ”Cross-Reference List”

All the inputs, outputs, markers, ... which you used are listed in the cross-reference list and the locations where they occur in the program are specified.
You can output either the global, i.e. interblock, cross-reference list or the local, i.e. blockwise, cross-reference list.

The following functions are provided in this pulldown menu:

```
Print Cross Reference List
Start Print
Symbols and Comments    On
Title Block             On
Title Block File        TITLE.DOC
Output filename:        STATION.LST
Cross Reference Mode    Local (Blockwise)
Signal List:            *
Block List:             OB1
Load Block List
Store Block List
Page Mode               blockwise
Start Page Number:      1
```

Cross-Reference Mode

- ”Print”, ”Cross-Reference List”, ”Cross-Reference Mode”

Here you must decide whether you want to output the global, i.e. interblock, cross-reference list or the local, i.e. blockwise, cross reference list. You can toggle between ”global” and ”local”.

local:	the cross-reference list affects only the specified block, e.g. I2.1 5(I)
global:	the cross-reference list affects the specified block and all blocks called by this block, e.g. I2.1 FB1 5(I)

In the ”block list” you must define the blocks from which the cross-reference lists should be created. An ”I” or ”O” appears after the numbers of the networks in which the corresponding signal occurs as input or output.

4.6.6 Signal Occupancy List

- "Print", "Signal Occupancy List"

All the inputs, outputs, markers, ... which you used and which are allocated in the blocks of the "block list" you created are listed in the Signal Occupancy List.

An "X" at the corresponding location of the signal table means that this signal occurs in the blocks entered in the "block list".

It is possible to output either the global, i.e. interblock Signal Occupancy List or the local, i.e. blockwise Signal Occupancy List.

The following functions are provided in this pulldown menu:

```
Print Signal Occupancy List
Start Print
Title Block               On
Title Block File          TITLE.DOC
Output filename:          STATION.LST
Occupancy Mode            Local (Blockwise)
Signal List:              *
Block List:               OB1
Load Block List
Store Block List
Page Mode                 blockwise
Start Page Number:        1
```

Occupancy Mode

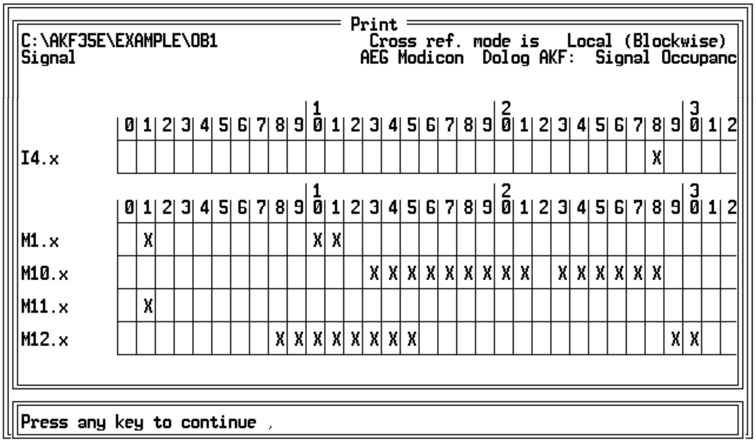
- "Print", "Signal Occupancy List", "Occupancy Mode"

You must decide here whether you want to output the global, i.e. interblock Signal Occupancy List or the local, i.e. blockwise Signal Occupancy List. You can toggle between "global" and "local".

local: the allocation list affects only the specified block
global: the allocation list affects the specified block and all blocks called by it

You must decide in the "block list" from which blocks the signal allcoation lists should be output. An "X" appears after the numbers of the networks in which the corresponding signal occurs as output.

The following figure shows an example of the Signal Occupancy List on the monitor.



4.6.7 Station Setup

- ”Print”, ”Station Setup”

All the data belonging to the initial setting are printed with the function Print Setup Station.

The following functions can be selected and called in this pulldown menu:

```
Print Station Setup
Start Print
Title Block                Off
Title Block File           TITLE.DOC
Output Unit                Monitor
Start Page Number:         1
```

The definitions contained in the following figure belong to the setup station:

```
PC* Station Presetting

    Plant                                  :  C:\AKF35E\
    Station                                :  EXERCISE
    Networking mode                        :  RS232
    Transmission Rate                      :  9600
    ALU Type                               :  ALU 021
    Part no. of the PC* basic software     :  275147/01
    Address Mode                           :  DIN
    Addressing                             :  SYM
    Max. number of Blocks                  :  250

PC* - Initial Parameterization

    AKF-RAM- / EPROM-Version               :  RAM
    AKF-EPROM-Segment-No                   :
    AKF-RAM-Segment-No                     :  5,6,7,8
    RAMZU-PUTE reserved                    :  Yes
    RAMZU-SEAB reserved                    :  No
    BSW-Segments                           :  17,18,19,20,21,22,23,24
    Reserved Segments                      :  9,10,11,12,13,14
```

The Dolog SFB address table is also output for basic software < version 6.0 , possibly extended by definitions from existing user blocks.

4.6.8 Command File

- "Print", "Command file"

The command file created under "Edit" is processed here. The lists contained in the file are output in the formats specified there. Several files can be printed with one command.

Note Recommendation: Look at the command file on the monitor as a check before printing.

You can select and call the following functions in this pulldown menu:

```
Execute Command File
Start Print
Page Mode               blockwise
Start Page Number:      1
Command File Name:      STD      .KMD
Output Unit             Monitor
```

4.6.9 Signal Memory Contents

- "Print", "Signal Memory Contents"

The file loaded by the PLC is output here with the signal memory contents. In this way archived memory contents can be analyzed at a later date or special initial machine settings can be documented.

You can select and call the followng functions in this pulldown menu:

```
Print signal memory content
Start Print
Title Block                   Off
Title Block File              TITLE.DOC
Output Unit                   Monitor
File Name     :               STATION  .SSP
Start Page Number:            1
```

4.6.10 Data Structure

- ”Print”, ”Data Structure”

You can print all the data structures existing in the data structure editor under ”Edit” here (AEG and user data structures).

You can select and call the following functions in this pulldown menu:

```
Print data structures
Start print
Title Block          Off
Title Block File     TITLE.DOC
Output Unit          Monitor
Start Page Number:   1
```

The following figure shows one page of the printout.

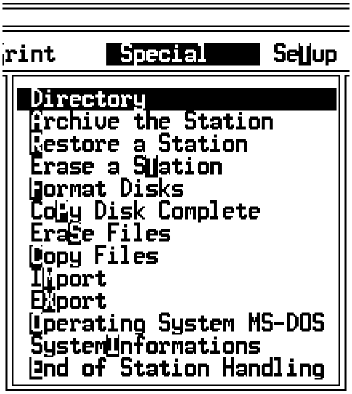
```
                                   Print
C:\AKF35E\EXAMPLE      AEG Modicon  Dolog AKF : Data Structures

Data struct. name : <I      > Ext. load.: No      Init at start: No
Samples           : 160       SYM/COM   : Yes     Onl. exchange: Yes
                              EQL-List  : Yes     Initial val. : No

Number |element type| Attribute              | Type   | Comment
       |            | read  | write | Sys-wr |        |
-----------------------------------------------------------------
   032 |  Bit       | Yes   | No    | Yes    |  SDC   |
-----------------------------------------------------------------


Data struct. name : <BESLIS> Ext. load.: No      Init at start: No
Samples           : 1        SYM/COM   : No      Onl. exchange: Yes
                             EQL-List  : No      Initial val. : Yes

Number |element type| Attribute              | Type   | Comment
```

4.7 Special

The functions for the programming panel are included in this menu.

Note If the message ”Command and file name unknown” appears in the following functions, make sure that your system/station directory or your path specification is complete and correct (see also MS-DOS manual).

The following system functions are provided:



4.7.1 Directory

- "Special", "Directory"

You can display the table of contents of your current station with this function. You then obtain a survey of the AKF blocks or DOS files contained there.

```
Directory
AKF-Blocks
DOS-Files
```

Directory of AKF blocks

- "Special", "Directory", "AKF Blocks"

A pulldown menu with the following functions appears:

```
Directory
Start Output
Search (Mask) :
Output Mode   :         Long form
Title Block             Off
Title Block File        TITLE.DOC
Output Unit             Monitor
```

Start Output

- "Special", "Directory", "AKF Blocks", "Start Output"
- "Special", "Directory", "DOS Files", "Start Output"

The output can be started here when the remaining parameters have been defined.

Search for (mask) AKF blocks

- "Special", "Directory", "AKF Blocks", "Search (Mask)"

Enter with the line editor a mask for the files to be displayed:

e.g.:	Empty line:	all blocks are displayed
	FB*	Display all function blocks
	PB2,FB*	Display PB2, then all FBs

Output mode of AKF blocks

- "Special", "Directory", "AKF Blocks", "Output Mode"

You can define how to output the table of contents with this function.
You can toggle between:
long form / short form.

Long form means:	all the data of the file is output (length, date, etc.). The files are sorted alphabetically.
Short form means:	only the file names are output. The files are sorted alphabetically.

Directory DOS files

- "Special", "Directory", "DOS Files"

A pulldown menu with the following functions appears:

```
Directory
Start Output
Search (Mask) :        A:                          *
Output Mode   :        Unsorted long form          *
Title Block            Off
Title Block File       TITLE.DOC
Output Unit            Monitor
```

* here difference from AKF blocks

Search for (mask) DOS files

- "Special", "Directory", "DOS Files", "Search (Mask)"

Using the line editor, enter a mask for the files to be displayed:

e.g.	*.*	Display all files
	*.txt	Display all text files

Output mode DOS files

- "Special", "Directory", "DOS Files", "Output Mode"

You can define how the table of contents is to be output with this function. You can toggle between:

Sorted long form / Unsorted long form
Sorted short form / Unsorted short form.

Long form means:	All the data of the files is output (length, date, etc.). The files are "sorted" in alphabetical order.
Short form means:	Only the file names are output. The files are "sorted" in alphabetical order.

4.7.2 Archive the Station

- ”Special”, ”Archive the Station”

All the data of a station can be saved on diskette with this function.

Caution

- **All the ”old” files on the diskette are deleted during archiving.**
- **When archiving, do not forget to label your diskettes with the complete station name and diskette number. You can only restore with the same station name.**

The following functions are provided:

```
Archiv
Start Archiv
From Station: C:\AKF35\A\*.*
To Station:   A:
```

Start Archiving

- ”Special”, ”Archive the Station”, ”Start Archiv”

The archiving is started after entry of the station to be archived and the target system.

Now follow the commands on the screen.

Caution   All ”old” files on the diskette are deleted during archiving.

The archiving can only be interrupted by simultaneously pressing the keys <Ctrl>+<C>.

After archiving you immediately return to the menu.

From Station

- ”Special”, ”Archive the Station”, ”From Station”

The station (parts) to be archived are entered here using the typewriter keyboard (line editor):

e.g.: C:\AKF35\EXAMPLE*.*

Step 1   <Return> (Start edit)

Step 2   Enter station name

Step 3   Terminate with <Return>

To Station

- ”Special”, ”Archive the Station”, ”To Station”

The target drive is entered here using the typewriter keyboard (line editor).

Entries ”A:” and ”B:” make sense here.

Step 1 <Return> (Start edit)

Step 2 Enter target drive

Step 3 Terminate with <Return>

4.7.3 Restore a Station

- "Special", "Restore a Station"

This function restores again the stations previously archived under the same name as during archiving.

Warning Existing files with the same name are overwritten without acknowledgement request!

The following functions are provided:

```
Restore
Start Restore
From Station: A:
To Station:   C:\AKF35\A\*.*
```

Start Restoring

- "Special", "Restore a Station", "Start Restore"

The restore is started with this function after definition of the source drive and the station to be restored.

Now follow the commands on the screen.

Caution The name of the target system must be the name with which the station was archived.

Restoring can only be interrupted by simultaneously pressing the keys <Ctrl>+<C>.

You immediately return to the menu after restoring.

From Station

- "Special", "Restore a Station", "From Station"

The source drive is entered here using the typewriter keyboard (line editor).

The entries "A:" and "B:" make sense here.

Step 1 <Return> (Start edit)

Step 2 Enter source drive

Step 3 Terminate with <Return>

To Station

- "Special", "Restore a Station", "To station"

Enter the target system (with path) using the typewriter keyboard (line editor):

e.g.: C:\AKF35\EXAMPLE*.*

The name must agree with the name during archiving.

STOP **Warning Existing files with the same name are overwritten!**

Step 1 <Return> (Start edit)

Step 2 Enter station name

Step 3 Terminate with <Return>

4.7.4 Erase a Station

- ”Special”, ”Erase a Station”

The specified station is deleted irrevocably after acknowledgement request. You can display a selection window after entering a space and <Return>.

Note The complete station name must be specified for this function, e.g. ”C:\AKF35\EXAMPLE” and the station to be deleted may not be the same as the current station processing.

4.7.5 Format Disks

- ”Special”, ”Format Disks”

New commercial diskettes must first be formatted before they can be used as a data carrier. Formatting prepares the diskette so that data can be stored and read by the MS-DOS operating system.

Caution All the data on the data carrier are destroyed during formatting (can also be used to completely delete a diskette).

The following functions are provided in this pulldown menu:

```
Format (Disks)
Start Formatting
Drive:        A:
Parameter:    /V
```

Start Formatting

- ”Special”, ”Format Disks”, ”Start Formatting”

Formatting is started after specification of the drive and possibly parameters.

Follow the commands on the screen after selecting the function.

Formatting cannot be aborted.

Drive

- ”Special”, ”Format Disks”, ”Drive”

The data carrier to be formatted is entered here by toggling.

A: Diskette in drive A: is formatted

Parameters

- ”Special”, ”Format Disks”, ”Parameter”

You can define the additional parameters for formatting here using the line editor (additional information can be found in the MS-DOS manual).

Parameters:
/S Copies the system files to the new data carrier
/1 Formats the diskette on one side
/8 Formats the diskette with 8 sectors per track
/V Permits you to assign a name to the data carrier (max. 11 characters including spaces)
/B Leaves room on the diskette for the operating system
/4 Formats a double-sided diskette with 360 kbyte capacity on a drive with high capacity. Such a diskette cannot be read reliably on 360 kbyte drives.
/3 Formats a double-sided 3.5” diskette with 720 kbyte capacity on a drive with high capacity. Such a diskette cannot be read reliably on 720 kbyte drives.

4.7.6 Copy Disk Complete

- ”Special”, ”Copy Disk Complete”

The complete contents of a diskette are copied to another diskette (same capacity) with this function. The data on the target diskette can be destroyed since the target diskette is formatted during copying.

The following functions are provided in this pulldown menu:

Start diskette copy

- ”Special”, ”Copy Disk Complete”, ”Start Copying Disk”

The copy is started after definition of the source and target drives.

Follow the commands on the screen after selecting the function.

This function can be aborted with <Ctrl>+<C>.

Source drive or target drive

- ”Special”, ”Copy Disk Complete”, ”Source Drive” or ”Target Drive”

You can define the same drive twice or different drives with the line editor.

Ex.1, one drive:	Ex.2, two drives:
Source drive: A:	Source drive: A:
Target drive: A:	Target drive: B:
	- or B to A.

4.7.7 Erase Files

- ”Special”, ”Erase Files”

You can delete any number of AKF blocks in the current station or DOS files with this function.

Erase AKF blocks

- ”Special”, ”Erase Files”, ”AKF Blocks”

A mask is defined for the block(s) to be deleted using the line editor. You can display a selection window after entering a space and <Return>. This function is only valid for the current station with AKF blocks.

e.g.	*.*	Delete all files
	FB	Delete all FBs

The delete can only be interrupted by simultaneously pressing the keys <Ctrl>+<C>.

Erase DOS files

- ”Special”, ”Erase Files”, ”DOS Files”

A mask for the file(s) to be deleted is specified with the line editor.

e.g.	*.*	Delete all files
	C:\AKF35\BSPANL*.TXT	deletes on the hard disk in the station ”BSPANL” all files with the extension .TXT

The delete can only be interrupted by simultaneously pressing the keys <Ctrl>+<C>.

4.7.8 Copy of Files

- "Special", "Copy Files"

You can load AKF blocks of the current station or DOS files to or from diskette with this command. AKF blocks of the current station or DOS files on hard disk can also be copied from one station to another.

Copy of AKF Blocks

- "Special", "Copy Files", "AKF Blocks"

The following functions are provided in this pulldown menu:

```
AKF
Start Copying
Source:                  PB1
Target:                  PB1
```

You can store a block under another name (i.e. copy) with this function.

Start file copy

- "Special", "Copy Files", "AKF Blocks", "Start Copying"
- "Special", "Copy Files", "DOS Files", "Start Copying"

The copy is started after definition of the source, target and possibly parameters.

Source

- "Special", "Copy Files", "AKF Blocks", "Source"

The block to be copied is defined (with the line editor) as source. You can display a selection window after entering a space and <Return>.

If only a block name and not a path is entered, the block of the current station is copied.

Target

- "Special", "Copy", "AKF Blocks", "Target"

The target block (block name under which the copied block is to be stored) is specified here. You can display a selection window after entering a space and <Return>.

Target e.g.:	A:	to diskette
	C:\TEST	to hard disk under directory "TEST"
	C:\TEST\FB10	the source is copied to "FB10" on hard disk under directory "TEST".
	no specification	to the current station

Copy of DOS files

- ”Special”, ”Copy Files”, ”DOS Files”

The following functions are provided under this pulldown menu:

```
DOS
Start Copying
Source:       *.*
Target:       A:
Parameter:    /V
```

*

* here difference to AKF blocks

You can use the replacement characters (”Wildcards”) ”?” and ”*” in the names of the source and target.

When using replacement characters, make sure that you do not accidentally overwrite important files.

Source

- ”Special”, ”Copy Files”, ”DOS Files”, ”Source”

The file to be copied with the complete path is specified (with the line editor) under source.

e.g.:	C:\AKF35\EXAMPLE\GSW.SPS	File GSW.SPS of the station ”EXAMPLE” in directory AKF35 on hard disk is copied to ”target”.

Target

- "Special", "Copy Files", "DOS Files", "Target"

The target directory for the file(s) included under "Source" is defined here with the line editor.

Target e.g.:	A:	on diskette
	C:\TEST	on hard disk under directory "TEST"
	no specification	in the current station

Parameters

- "Special", "Copy Files", "DOS Files", "Parameters"

You can specify additional parameters to copy here using the line editor (additional information can be found in the MS-DOS manual).

Parameters:

/V: Check of the copy

/A: The file is treated as text file.
Source: Copy up to the EOF character (exclusive).
Target: The file is terminated with the EOF character.

/B: Source: The whole file is copied.
Target: The file is not terminated with the EOF character.

4.7.9 Import

- "Special", "Import"

Data structures, blocks and symbols and comments can be imported from import files (i.e. inserted in the current station) with this function. In order to be able to import with this function, you must previously have made an "export" in another station or generated an ASCII import file with any editor (only for SYM/KOM).

The following functions are provided:

```
Import
Data Structure
Blocks
Symbols and Comments
```

Import data structure

- "Special", "Import", "Data Structure"

The data structures can be read in from the entered import file (data base) here. The data structures are subject to the same conditions as in the data structure editor and are integrated in the defined station.

The following functions are provided:

```
Import
Start
Source Directory :
Import-Filename  :  EXPORT  .EXP
Structure List   :  *
```

Start Importing

- ”Special”, ”Import”, ”Data Structures”, ”Start”
- ”Special”, ”Import”, ”Blocks”, ”From Export File”, ”Start”
- ”Special”, ”Import”, ”Blocks”, ”From Station”, ”Start”
- ”Special”, ”Import”, ”Blocks”, ”From DOS File”, ”Start”
- ”Special”, ”Import”, ”Symbols and Comments”, ”Start”

The data to be imported from the entered import file is transferred to the defined station.

Import source directory

- ”Special”, ”Import”, ”Data Structures”, ”Source Directory””
- ”Special”, ”Import”, ”Blocks”, ”From Export File”, ”Source Directory”
- ”Special”, ”Import”, ”Blocks”, ”From Station”, ”Source Directory”
- ”Special”, ”Import”, ”Blocks”, ”From DOS File”, ”Source Directory”
- ”Special”, ”Import”, ”Symbols and Comments”, ”Source Directory”

The complete path containing the import file is entered here.

If no path is entered, the import file is expected in the current station directory.

Importing import file name

- "Special", "Import", "Data Structures", "Import File Name"
- "Special", "Import", "Blocks", "From Export File", "Import File Name"
- "Special", "Import", "Symbols and Comments", "Import File Name"

The name of the file to be imported is entered here.

- Data Structures/Blocks
  File name: EXPORTnn.EXP (nn = 01 ... 99)

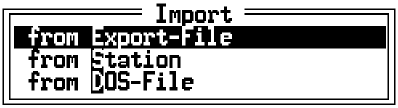
- Symbols and Comments
  File name xxxxxx.ASD/FSD (FSD: error file, see page 300)

If there is no input, a list of all the import files existing in the source directory which are suitable for the selected function is generated. An import file can then be selected from this list.

Importing blocks

- "Special", "Import", "Blocks"

Blocks can be read in here from the following sources:



Importing blocks from export file

- "Special", "Import", "Blocks", "From Export File"

You can import blocks from a data base with this function

```
==================== Import ====================
 Start
 Source Directory :
 Import-Filename  :  EXPORT   .EXP
 Block List       :  *
```

Importing blocks - block list

- "Special", "Import", "Blocks", "From Export File", "Block List"
- "Special", "Import", "Blocks", "From Station", "Block List"
- "Special", "Import", "Blocks", "From DOS File", "Block List"

Define the blocks to be imported from the import file here.

Importing blocks from station

- "Special", "Import", "Blocks", "From Station"

You can import blocks from a specified source station to the current station with this function.

```
==================== Import ====================
 Start
 Source Station   :
 Block List       :  *
```

Importing blocks from DOS file

- ”Special”, ”Import”, ”Blocks”, ”From DOS File”

You can import blocks which were previously stored as a DOS file from a specified source directory with this function

```
═══════════ Import ═══════════
 Start
 Source Directory :
 Block List       :   *
```

Importing symbols and comments

- ”Special”, ”Import”, ”Symbols and Comments”

The symbols, comments and initial values can be read in from an import file here.

The following functions are provided:

```
══════════════Import══════════
 Start
 Source Directory: C:\AEG-A91\AKF35\SYMKOM.ASD
 Import File Name: REGELN.ASD
```

Note For **closed-loop control**:
import the file REGELN.ASD.
For intelligent function modules **POS 102 or POS 112**:
import the file POS.ASD.

The symbols and initial values are subject to the same conditions as in the symbol and comments editor and are integrated in the system.

If an error occurs during reading, an error file with the name of the import file but with the extension '.FSD' is set up in the current directory (of the station). This file can be corrected with an ASCII editor and then be read in again.

Create ASCII file

In order to ensure correct execution of the function, you must satisfy the following conditions when creating the ASCII file:

The symbol may not be used for another signal in the SYM/COM block.

The operand must also be valid with regard to the equipment list.

The initial value must be valid for the data width and the display mode of the operand.

Each line begins with a ':' followed by a reference letter. The following reference letters are allowed:

I,Q → Info lines
Z → Line comment
S → Signal line
K → Comment line
F,W → Event lines (error or warning text)

Lines not beginning with a colon are ignored without a comment. All subsequent lines in brackets [...] are optional.

❐ **Info lines**

[:I AKF Type Version]	This line contains the AKF type and version.
[:Q System,Station,Date]	This line contains the system name, station and date of creation.

These lines are created by the export function and are of no relevance for importing.

❐ Operand line (Signal line)

:S Operand,[Symbol],[Display Mode],[Initial Value]

☞ **Note** If the parameters specified here in brackets are not defined, they are automatically set to ”0” or overwritten with spaces. Values can therefore be overwritten in the station into which you import.

The operand must also be valid with regard to the equipment list.

The symbol may have up to 8 characters. It may not contain commas. Semicolons are converted into underlining. Spaces are removed. Symbols which are already defined are rejected.

The display mode and initial value must be logical and valid for the element type of the operand.

Valid display modes for element types:

Bit	BIN
Byte	BIN, SDC, DEC, HEX, OCT, CHR
Word	BIN, SDC, DEC, HEX, OCT, CHR
Double word	SDC, DEC, HEX, OCT, CHR
Floating word	FLP
Pointer	HEX

❐ Comment lines

[:K Comment]	The comment is any text of up to 40 characters.
[:Z Line comment]	The lINE comment can also be any text. Up to 100 lines of 60 characters each can be entered.

Texts which are too long are cut without a message.

An entry contains at least one operand line.

The line comment is assigned to the following operand line, the comment is assigned to the previous operand line. If there is no comment for an operand, the corresponding comment line may be omitted.

An entry is terminated with a semicolon in the first coloumn.

❐ Event lines

:F	Error message
:W	Warning message

These lines are stored in the error file if an error occurs during importing. If the import file is an error file, these lines are ignored.

Example

The following file was correctly created and contains no error.

```
:I System name:AKF35 Version 6.0
:Q System: C:\AKF35\, Station: TTT, Date: 15.12.1991
.
:S Q26.1,E01K11,BIN,–
:K Protection Roller track Forwards       Unstacker
;
:S Q26.2,E01K18,BIN,–
:K Protection Roller track Backwards      Unstacker
;
.
:Z **************************
:Z Marker for PB enable
:Z **************************
:S M1.1,FRGPB1,BIN
:K Enable variant PB TYP 1
;
:S M1.2,FRGPB2,BIN
:K Enable variant PB TYP 2
;
:S M1.3,FRGPB3,BIN
:K Enable variant PB TYP 3
;
.
:Z ********************
:Z Data structure KIPP
:Z ********************
:S KIPP1,KIPPEN,–,–
:K Internal variables FB Tipping belt Upper part
;
:S KIPP1.80,,SDZ
:K Previous state Status word 1
;
.
.
:S KIPP1.92,,SDZ,50
:K Suitcase conveying time in tipping belt
;
```

Lines 7, 17 and 29 are modified as follows for test purposes

Line 7 `:S Q26.2,E01K11,BIN,–`
Line 17 `:S M1.2,FRGPB2,DEC,5`
Line 29 `:S KIPP1.80,,DEC,12FF`

The error file created during importing is as follows (brackets are remarks):

```
:F Symbol already exists
:S Q26.2,E01K11,BIN,–                    (same symbol as A26.1)
:K Protection Roller track Backwards Unstacker
;
:F Display mode invalid
:S M1.2,FRGPB2,DEC,5           (Markers only have display mode BIN)
:K Enable variant PB TYP 2
;
:F Initial value invalid
:S KIPP1.80,,DEC,12FF   (Initial value incorrect because hexadecimal value)
:K Previous state Status word 1
;
```

4.7.10 Export

- "Special", "Export"

Data structures, blocks and symbols and comments can be exported (i.e. removed from the current station) with this function.

The following functions are provided:

```
═══════════ Export ═══════════
 Data Structure
 Blocks
 Symbols and Comments
```

Export data structures

- "Special", "Export", "Data Structure"

Data structures can be stored in an export file here.

The following functions are provided:

```
═════════════════ Export ═════════════════
 Start
 Target Directory :
 Export-Filename  :  EXPORT   .EXP
 Structure List   :
```

Start Export

- ”Special”, ”Export”, ”Data Structure”, ”Start”
- ”Special”, ”Export”, ”Blocks”, ”To Export File”, ”Start”
- ”Special”, ”Export”, ”Blocks”, ”To DOS File”, ”Start”
- ”Special”, ”Export”, ”Symbols and Comments”, ”Start”

The data to be exported is stored in the export file.

Export Target Directory

- ”Special”, ”Export”, ”Data Structure”, ”Target Directory””
- ”Special”, ”Export”, ”Blocks”, ”To Export File”, ”Target Directory”
- ”Special”, ”Export”, ”Blocks”, ”To DOS File”, ”Target Directory”
- ”Special”, ”Export”, ”Symbols and Comments”, ”Target Directory”

The complete path in which the export file is to be stored is entered here. If this path does not exist, it is set up when the function is started. If this is not possible, the function is aborted.

If no target directory is defined, the export file is stored in the current station directory.

The default value is the current station directory.

Export Export-File Name

- ”Special”, ”Export”, ”Data Structure”, ”Export-File Name”
- ”Special”, ”Export”, ”Blocks”, ”To Export-File”, ”Export-File Name”
- ”Special”, ”Export”, ”Symbols and Comments”, ”Export-File Name”

The name of the file in which the data is to be stored is entered here. Any name is permitted.

❒ Data Structures/Blocks
File name: EXPORTnn.EXP (nn = 01 ... 99)

❒ Symbols and Comments
File name xxxxxx.ASD

Export data structure / Structure list

- "Special", "Export", "Data Structure", "Structure list"

A list of data structure names may be entered. This list may contain up to 200 characters. The entries must be separated by commas.

If an empty field is entered, a list of all the user data structures of the defined station is generated. Individual data structure names can be selected from it to form a list.

Export blocks

- "Special", "Export", "Blocks"

Blocks can be stored in a data base or in DOS files here.

The following functions are provided:

```
Export
 to Export-File
 to DOS-File
```

Export blocks to export file

- "Special", "Export", "Blocks", "To Export-File"

Blocks can be stored in a data base of a target directory here.

The following functions are provided:

```
Export
Start
Target Directory :
Export-Filename  :  EXPORT  .EXP
Block List       :  *
```

Export blocks / Block list

- "Special", "Export", "Blocks", "To Export File, "Block List"
- "Special", "Export", "Blocks", "To DOS File, "Block List"

You define which blocks should be exported in this list, e.g. PB, FB10 ...

Export blocks to DOS file

- "Special", "Export", "Blocks", "To DOS File"

Blocks can be stored in DOS files of a target directory here.

The following functions are provided:

```
Export
Start
Target Directory :
Block List       :  *
```

Export symbols and comments

- "Special", "Export", "Symbols and Comments"

Symbols, comments and initial values can be stored in an export file here. The file must have the extension .ASD. In order to be able to import, you must have exported to another station previously or generated an ASCII import file with any editor (only for SYM/COM).

The following functions are provided:

```
Export
Start
Target Directory: C:\AKF35\A
Export File Name:         .ASD
```

4.7.11 Operating System MS-DOS

- ”Special”, ”Operating System MS-DOS ”

Excerpt from the MS-DOS manual:

”What is MS-DOS ?
MS-DOS stands for Microsoft Disk Operating System. It controls the basic functions of the computer and links the individual boards in such a way that user programs (word processing, calculation, data base, etc.) can easily make use of them. MS-DOS enables data to be stored in files on disk or hard disk and recalled, data to be entered from the keyboard and output again on a printer.
MS-DOS lets you copy, delete, compare, rename, save data from disk. It creates directories of your data media and provides every entry in them with time and date. It also controls the saving of your data from hard disk to a tape drive.”

You can now execute the functions described in the MS-DOS manual. You return to the Dolog AKF software by entering ”EXIT”.

4.7.12 System Information

- ”Special”, ”SystemInformations”

This function informs you about the hardware configuration of your PADT, e.g. about the processor, the DOS version, the interfaces and about main memory. Output begins after the following menu:

```
                 Print system information
 Start print
 Title Block          Off
 Title Block File     TITLE.DOC
 Output unit:         Monitor
```

4.7.13 End of Station Handling

- ”Special”, ”End of Station Handling”

You terminate processing of the Dolog AKF software for the A350/A500 with this function.

You can now start the AKF software again with another call.

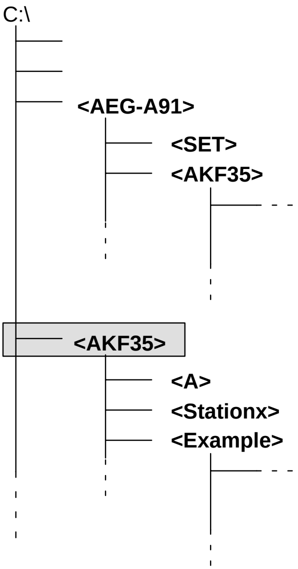
4.8 Setup

The Setup functions are for presetting fixed data of the system. This includes the following settings:

The menu option ”Station” should always be processed before editing the user program.

Directory structure on the PADT

The Dolog AKF software is installed in two different directories on the hard disk.



– to the program path (presetting ”C:\AEG-A91”)
Any path must be defined during installation.

– to the data path (presetting ”C:\AKF35”)
It is divided into the plant and the station path.

Mark: This is the data path of the example plant.
Any path must be defined during installation.

4.8.1 Plant

- ”Setup”, ”Plant”

A plant corresponds to the directory on the PADT. It can contain a number of different stations. Each station is a further directory.

The plants are set with the following function.

If the entered plant does not yet exist, it is generated after inquiry.

The last processed plant is automatically set again for existing plants.

Example: C:\AKF35

4.8.2 Station

- "Setup", "Station"

This function presets values for the station generated by the user:

```
PC* Station Presetting
PC* Station Name            A
ALU Type                    ALU 150
Address Mode                DIN
Addressing                  ABS
Max. Number of Blocks:      250
Link Mode                   Complete Retranslation
Input Mode                  LD
First Use of PC*
Setup PC* Data (SYRES)
Read PC* Data (SYKON)
```

PLC Station Name

- "Setup", "Station", "PC* Station Name"

You can select or create a station using the line editor (typewriter keyboard) with this function. If you enter a space, a window with all existing stations of the current plant is displayed. A station can be selected from this window using the cursor keys and <Return>.

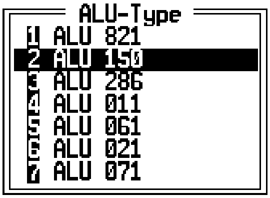
If the entered station does not exist, it is generated with the <Return> key after acknowledgement (corresponds to the directory structure of the PADT).

All subsequent processing functions are then executed in this station and the generated files are stored in this directory.

ALU Type

- ”Setup”, ”Station”, ”ALU Type”

When this function is selected, a window from which you can select the required controller (ALU) is opened:



Note The configuration of the loadable basic software version 6.0 is only possible with ALU 021 and ALU 071. The function ”Dolog SFB–Address Table” is no longer required.

Max. Number of Blocks

- ”Setup”, ”Station”, ”Max. Number of Blocks”

You can define how many blocks (10 ... 1 999) are to be loaded into the programmable controller here. Dolog AKF then reserves the appropriate space when linking your program. If the defined amount is too small, it is not possible to link the program. You should enter the number of blocks (appears as a message) with an additional 10 - 20% during the last link.

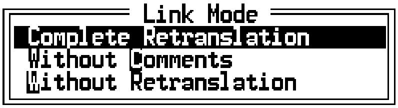
Note No new block can be added online if the list is full.

Link Mode

- ”Setup”, ”Station”, ”Link Mode”

The link mode defines which additional information about the user program is to be stored in the programmable controller.

You can choose between:



Complete Retranslation (Representation)

- ”Setup”, ”Station”, ”Link Mode”, ”Complete Retranslation”

All the line and network comments and the network titles are linked to the program with this setting. The space requirements in the programmable controller are displayed at the end of the link procedure.

Note Symbols and comments from ”Edit”, ”Symbols and Comments” are not transferred to the programmable controller and therefore cannot be represented. The initial values of the SYM/COM block are saved in the initial value block and transferred to the PLC. This block is also read out during retranslation.

Without Comments

- ”Setup”, ”Station”, ”Link Mode”, ”Without Comments”

The program can be represented without any comments with this setting. Line/ network comments and network titles can no longer be represented.

Without Retranslation

- "Setup", "Station", "Link Mode", "Without Retranslation"

Representation from the PLC is no longer possible with this setting. The load function "Read Out PC*" is therefore inapplicable.

Input Mode

- "Setup", "Station", "Input Mode"

You can toggle between instruction list (IL), ladder diagram (LD) and function block diagram (FBD). A short description of the individual expert languages can be found in Part V, chapter 3.

First Use of PLC

- "Setup", "Station", "First Use of PC*"

Caution   This function is only possible with RS232 connections.

With this function, the first-time parametrization of the programmable controller is carried out with a command after specification of some parameters.

Functions which are valid for all ALU types:
The following functions are provided by "First Use of PC*":

```
                              First Use of PC*
 Start Function
 RAM/EPROM Version          : RAM
 EPROM-Segment numbers      :
 RAM-Segment numbers        : 5,6,7
 Set up RAMZU-PADT          : Yes
 Set up RAMZU-SEAB          : No
 Reserved Segment numbers   : 8,9,10,11,12,13,14,15
```

* For ALU 021 / ALU 071 see also page 320

Start Function

- ”Setup”, ”Station”, ”First Use of PC*”, ”Start Function”

The first-time parametrization of the progammable controller is started with this function after setting the remaining parameters and some safety inquiries. The PLC is set up and assigned initial values.

Required input:	RAM or EPROM	
	RAMZU-PADT	”yes”
	RAMZU-SEAB	”yes” for Modnet 1/SFB or Modnet 2/NP
	Options:	All other functions can be assigned a space. The entry is made automatically in the unused segments. The software makes a syntax check (see also ”Reserved Segments”).

An overview is displayed if the parametrization was successful.

Example for any PLC configuration with ALU 021:

PC* occupancy for : 275147/01

Memory area	Segment	Usage
1-4	5-8	AKF/RAM
31	15	RAMZU-PADT
32	16	SYKON

RAM/EPROM Version

- ”Setup”, ”Station”, ”First Use of PC*”, ”RAM/EPROM Version”

This entry defines whether your AKF program should execute in the RAM or EPROM. For program execution in write-protected RAM, see method in chapter 4.4.14.

The default values are on RAM.

☞ **Note** An online exchange is no longer possible for the EPROM version. The ”Program to PC*” function must be executed nevertheless in order to transfer the equipment list and initial values.

EPROM Segment Numbers

- ”Setup”, ”Station”, ”First Use of PC*”, ”EPROM-Segment Numbers”

If your program is stored in the EPROM, the segments of the EPROM area are entered here. If this is the case, at least one segment must also be entered in ”RAM Segment Numbers”.

There is no default value here (≙ no reservation).

RAM Segment Numbers

- ”Setup”, ”Station”, ”First Use of PC*”, ”RAM-Segment Numbers”

If your program is to execute in the RAM, the segment numbers are entered here. If nothing or a space is entered here, the segments are automatically entered when the first-time parametrization function executes. The remaining RAM is assigned to the ”Reserved Segment Numbers”.

Set up RAMZU-PADT

- ”Setup”, ”Station”, ”First Use of PC*”, ”Set up RAMZU-PADT”

The memory requirements for operation with the PADT are set up here.
The assignment is mandatory for the dynamic status display. The memory is entered automatically if ”yes” is set.

Note One segment must be provided for RAMZU-PADT and RAMZU-SEAB together. This segment may not be entered in ”RAM Segment Numbers”, ”Reserved Segment Numbers” or ”BSW Segments”. RAMZU-PADT requires 10 Kbytes, RAMZU-SEAB requires 4 Kbytes. You can use the remaining 18 Kbytes of the segment at will.

Set up RAMZU-SEAB

- ”Setup”, ”Station”, ”First Use of PC*”, ”Set up RAMZU-SEAB”

The memory requirements for a Modnet link are set up here.
The assignment is mandatory for the Modnet 1/SFB and Modnet 2/NP link. The memory is entered automatically if ”yes” is set.

Reserved Segment Numbers

- ”Setup”, ”Station”, ”First Use of PC*”, ”Reserved Segment Numbers”

Here the user can define segments to be used for other purposes and which may not be used for Dolog AKF (e.g. for Dolog B programs in the same PLC).

Additional Functions for ALU 021 and ALU 071

* same functions as for ALU 821, ALU 150, ALU 286, ALU 011, ALU 061

```
BSW-Segments          :
BSW-Configuration
```

BSW Segments

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW Segments”

The segments for the loadable basic software are reserved in this function (1 segment = 32,768 bytes, the module size is defined in the BSW editor). Make sure that you always reserve sufficient segments for the basic software (≥ version 6.0).

Segments 5 ... 24 (except for 16) can be entered. The entries should be sequential if possible.

BSW-Configuration

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”

You start the editor in which you set up your basic software with this function.

Requirement: ”ALU type”: ALU 021 or ALU 071

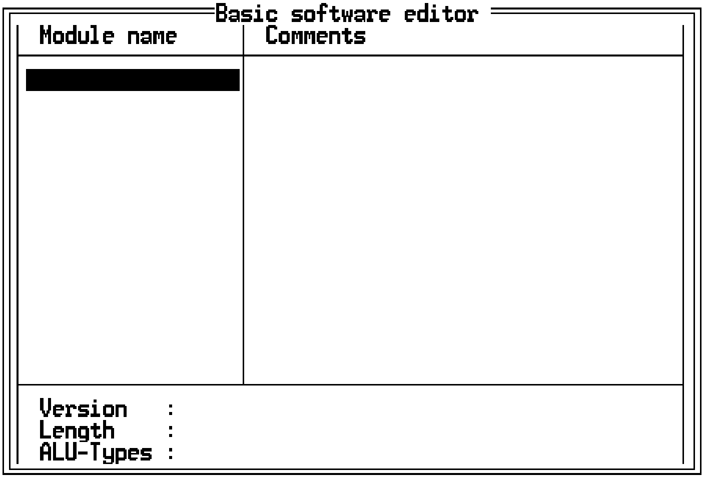
Caution Information about the contents of the individual modules can be found in the documentation of basic software version 6.0:
[6]A500
Grundsoftware Version 6.0
Benutzerhandbuch
A91M.12-279344

6) in german language

Basic Software Editor (Module List)

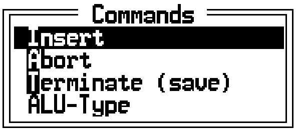
- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>

A window appears for module input after selection of this function.



The modules are now selected in the left column. The comment can be changed in the right column.

The following input is possible with <Return> in the column ”module name” :



ALU type cannot be selected

Insert Module / SFB

- "Setup", "Station", "First Use of PC*", "BSW-Configuration" <Return>, "Insert"

You can now decide whether to select modules or SFBs.

- If you need a certain SFB but do not know in which module it is, choose "SFB". A list of all SFBs available in the loadable BSW is displayed. The correct module is automatically entered in the module list when a SFB is selected.

Caution Information about the contents of the individual modules can be found in the documentation of basic software version 6.0:
[7]A500
Grundsoftware Version 6.0
Benutzerhandbuch
A91M.12-279344

Note The following SFBs are not elements of the loadable BSW and are automatically linked by the link program:

for intelligent function modules

VIP+	SFB1	EINR_POS	SFB62
VIPS+	SFB2	HAND_POS	SFB63
ISTD_POS	SFB61	AUTO_POS	SFB64

for closed-loop control

KPID	SFB300	PT2	SFB331
PID	SFB302	DT1	SFB335
PIDP	SFB304	PDT1	SFB340
PI	SFB308	IB	SFB345
ZR	SFB310	AB1	SFB350
DR	SFB315	AB2	SFB351
PBM	SFB320	TZ	SFB355
PDM	SFB325	STOE	SFB360
PT1	SFB330	O–REG	SFB390

7) in german language

❐ If you know exactly which modules of the software you need, you can select directly with ”Module”.

In general, each module can only be entered exactly once in the module list.

```
Basic software editor
 Module name        | Comments
--------------------+-------------------------------------
 ERGBAU             | additional modules
--------------------+-------------------------------------
 Version   : 1.1
 Length    : 10240
 ALU-Types : 021,071
```

After entering the first module, the following menu is provided:

```
Commands
Insert
Delete
Print
Info
Scroll Forwards
Scroll Backwards
Abort
Normalize
Terminate (save)
Sort
ALU-Type
Link(GSW.SPS)
Load BSW to PC*
```

cannot be selected

Insert

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”Insert”, Select Module or SFB, <Return>, ”Insert”

You can insert further modules/SFBs in the module list with this function.

A list of the modules can be found in chapter 2.10.1.

Delete

- ”Setup”, ”Station”, ”First Use of PC*”, ”Configure BSW >=V6.0”, <Return>, ”Insert”, Select Module or SFB, <Return>, ”Delete”

You can delete modules from the module list with this function. The module to which the cursor points is deleted.

Print

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”Insert”, Select Module or SFB, <Return>, ”Print”

You can print the module list with this function. The following settings are possible:

```
Print BSW-List
Start Print
MoDus                    BSW in PADT
Modul(list)
Title Block              Off
Title Block File         TITLE.DOC
Output Unit              Monitor
Start Page Number        1
```

Print Mode

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”Insert”, Select Module or SFB, <Return>, ”Print”, ”Mode”

Here you can select the print of the

- basic software on the programmable controller (all modules on the PLC are printed); the PLC must be connected for this printout. (BSW in PC*)
- basic software on the PADT (module list is printed); this printout is made off-line.(BSW in PADT)

Module (list)

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”Insert”, Select Module or SFB, <Return>, ”Print”, ”Module (list)”

Enter the modules to be printed here. Individual modules are separated by commas. All the modules are printed if you enter ”*” or an empty line.

This entry is ignored for ”BSW in PC*”. All the modules on the PLC are then printed.

You can call the following functions directly in the editor (outside the pulldown menu) with <Ctrl>+<reference character>.

Info

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”Insert”, Select Module or SFB, <Return>, ”Info”

The SFBs related to the selected module names are output with this function. You can page in the SFB list with <↓>, <↑>, <PgDn> or <PgUp>.

Scroll Forwards

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”Insert”, Select Module or SFB, <Return>, ”Scroll Forwards”

You can look at the previous page of the module list with this function.

You can page to the previous page outside of the pulldown menu with <PgUp>.

Scroll Backwards

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”Insert”, Select Module or SFB, <Return>, ”Scroll Backwards”

You can look at the next page of the module list with this function.

You can page to the next page outside the pulldown menu with <PgDn>.

Abort

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”Insert”, Select Module or SFB, <Return>, ”Abort”
- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”Abort”

With this function you leave the module list without storing it.

You can abort with <Esc> outside the pulldown menu.

Normalize

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”Insert”, Select Module or SFB, <Return>, ”Normalize”

The module list is deleted completely with this function. New entries are possible immediately afterwards.

Terminate (Save)

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”Insert”, Select Module or SFB, <Return>, ”Terminate (Save)”
- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”Terminate (Save)”

The module list is stored and the editor terminated with this function.

You can end with <F2> or <Ctrl>+<T> outside the pulldown menu.

Sort

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, Insert”, Select Module or SFB, <Return>, ”Sort”

The modules in the module list are sorted alphabetically with this function.

ALU Type

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”Insert”, Select Module or SFB, <Return>, ”ALU Type”
- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”ALU Type”

This function cannot be selected; it is reserved for a standalone–version of the loadable basic software.

Link Basic Software

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”Insert”, Select Module or SFB, <Return>, ”Link(GSW.SPS)”

The basic software is prepared for transfer to the programmable controller (linked) with this function. The file GSW.SPS is created.
The basic software can only be linked if the number of segments required to store the BSW was defined (1 segment = 32,768 bytes, the module size is displayed in the editor). (See function ”First Use of PC*”, ”BSW Segments”)

Load Basic Software in PLC

- ”Setup”, ”Station”, ”First Use of PC*”, ”BSW-Configuration”, <Return>, ”Insert”, Select Module or SFB, <Return>, ”Load BSW to PC*”

Note This function is only possible online.
An ”online exchange” (load/reload) of the loadable basic software modules is **not** possible.

This function transfers the basic software to the programable controller. This should only be done for the first time after the first-time parametrization of the PLC.

The basic software 6.0 can be loaded as follows:

- The basic software is transferred independently of the user program with this function; this makes sense if the user program will be extended at a later time but the modules required are already known.

- The basic software required in the user program is linked (GSW.SPS) and then transferred with the program to the PC using ”Load”, ”Program Link”.

Set up PLC Data (SYRES)

- ”Setup”, ”Station”, ”Set up PC* Data (SYRES)”

Caution This function is only possible for the RS232 link. Only a system backup (SYKON) which was already saved after the ”SSN” function may be restored (SYRES) after the BSDOL function ”SSN”. (see page 330)

The file generated with the function ”Read PC* Data” is transferred to the programmable controller and activated.

System variables are quickly restored in this manner. Furthermore, several PLC can be created with the same information (duplicated).

Read PLC Data (SYKON)

- "Setup", "Station", "Read PC* Data (SYKON)"

Caution This function is only possible with the RS232 link. You should only carry out the SYKON function after the BSDOL function "SSN".

The following functions are provided with this pulldown menu:

The system variables are saved in the assigned PLC segment with this function, i.e. archived (ALU 150/ALU 286/ALU 011/ALU 061/ALU 021/ALU 071: segment 16; ALU 821: segment 28).

The system backup can be loaded back into the PLC with the function "Set up PC* (SYRES)".

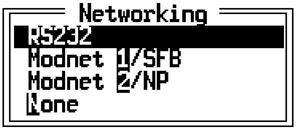
The system variables archived with this function are restored quickly in the PLC with "Set up PC* (SYRES)".
Several PLC can thus be set up with the same information (duplicated).
There is a defined starting behavior since certain defined bit/word areas can be archived and restored.

Note You should save the system backup after the program start in order to include the current starting address in the backup.

4.8.3 Networking

- ”Setup”, ”Networking”

The following types of networking are possible between the programmable controller and PADT:



You can select the required type of networking with the cursor keys and <Return>.

The window is closed and the setting activated with <Esc>.

RS232

- ”Setup”, ”Networking”, ”RS232”

Dolog AKF is configured using a RS 232 C or V.24 interface with this type of networking.

PADT and ALU are linked. After <Return>, the transmission rate is entered by toggling.

The default transmission rate is 9600 baud.

The particular transmission rate must agree with that of the PLC program interface:

A350: Define jumper on the SCU 150 and DIP switch on ALU 150.
A500: Set rotary switch on ALU 011/ALU 021,
DIP switch on ALU 061/ALU 071 (K1 ... K4),
jumper on UKA 024 and DIP switch on ALU 150/ ALU 821/
ALU 286.

Modnet 1/SFB

- "Setup", "Networking", "Modnet 1/SFB"

☞ **Note** Only Modnet 1/SFB **or** Modnet 2/NP can be driven. Simultaneous networking is not possible. Modnet 2/NP is only possible with A500.

You can program with Modnet 1/SFB if you satisfy the following requirements:

- Dolog AKF → A350/A500 ≥ version 5.0 installed on PADT, Modnet 1/SFB driver installed. The current hardware address setting must be entered during driver installation.
- PADT station number defined with COM → AKF
- RAMZU-SEAB in RS232 link set up with Dolog AKF
- System information configured with software COM → AKF
- A350 or A500 with basic software ≥ version 5.0
- BIK 001/002/003 available in PADT
- BIK available in programmable controller

BIK in the PADT and BIK in the PLC are connected with this type of networking.

☞ **Note** The following programming functions cannot be executed with this type of networking:
First Use of PC*, Terminal Mode, SYKON and SYRES.

Modnet 2/NP

- ”Setup”, ”Networking”, ”Modnet 2/NP”

Note Only Modnet 1/SFB **or** Modnet 2/NP can be driven. Simultaneous networking is not possible. Modnet 2/NP is only possible with A500.

You can program with Modnet 2/NP if you satisfy the following requirements:

- Dolog AKF → A350/A500 ≥ version 5.0 installed on PADT, Modnet 2/NP driver installed. The current hardware settings must be entered during driver installation.
- PADT station number defined with COM → AKF
- RAMZU-SEAB in RS232 link set up with Dolog AKF
- System information configured with software COM → AKF
- A500 with basic software ≥ version 5.0
- KP4 available in PADT
- KP1 available in A500

KP4 in the PADT and KP1 in the A500 are connected with this type of networking.

The drivers were already installed and the PADT station number defined during software installation.

Note The following programming functions cannot be executed with this type of networking:
First Use of PC*, Terminal Mode, SYKON and SYRES.

None

- ”Setup”, ”Networking”, ”None”

With the networking mode ”none” or if the software is called with the parameter /NOSPS, the mouse can be driven at the COM1 interface.

The setting ”none” is used for off-line programming of your user program.

4.8.4 Print

- "Setup", "Print"

The following initial values are possibile:

```
Print presetting
Output Unit          Monitor
Lines/Page:          66
Formfeed             On
```

Output Unit

- "Setup", "Print", "Output Unit"

The following are available as output unit:

Output Unit	
Monitor	Output on monitor
Printer	Printer selection appears here
File:	Output to a file

See also under "Print", "Program Log", "Output Unit" on page 183.

Printer

- "Setup", "Print", "Printer"

The menu provides some printer types. You can select the printer driven at your PADT here:

```
Printer Directory
A: Freely configurable RS232
B: DRU096
C: DRU120
D: DRU292
E: DRU293
F: DRU1200
G: Epson EX-800
H: Epson FX-80
I: NEC P60/P70
```

Observe the line length for DIN A4 printers with normal print: the signal comment may contain up to 32 characters. Otherwise you should set compressed print (optionally configurable).
These parameters can be found in the printer descriptions.

Freely Configurable RS232 Interface (Serial)

- "Setup", "Print", "Output Unit", "Printer", "Freely configurable RS 232"

The following parameters must be entered for a printer with freely configurable RS232 interface:

```
Printer initialisation
Serial Port (COMx):        COM1
Baudrate:                  9600
Parity:                    none
Data bits:                 8 Bit
Stop bits:                 1
Initialization:
NOrmal Letter:
NArrow Letter:             0F
Unstal Printer
```

DRU 096

- ”Setup”, ”Print”, ”Output Unit”, ”Printer”, ”DRU096”

You can select here between:

```
Port
A: Parallel Port 1 (LPT1)
B: Parallel Port 2 (LPT2)
```

DRU 120

- ”Setup”, ”Print”, ”Output Unit”, ”Printer”, ”DRU120”

You can select here between:

```
Port
A: Parallel Port 1 (LPT1)
B: Parallel Port 2 (LPT2)
C: Serial Port  1 (COM1)
D: Serial Port  2 (COM2)
```

DRU 292

- ”Setup”, ”Print”, ”Output Unit”, ”Printer”, ”DRU292”

You can select here between:

```
Port
A: Parallel Port 1 (LPT1)
B: Parallel Port 2 (LPT2)
C: Serial Port  1 (COM1)
D: Serial Port  2 (COM2)
```

DRU 293

- "Setup", "Print", "Output Unit", "Printer", "DRU293"

You can select here between:

```
Port
A: Parallel Port 1 (LPT1)
B: Parallel Port 2 (LPT2)
C: Serial Port  1 (COM1)
D: Serial Port  2 (COM2)
```

DRU 1200

- "Setup", "Print", "Output Unit", "Printer", "DRU1200"

You can select here between:

```
Port
A: Parallel Port 1 (LPT1)
B: Parallel Port 2 (LPT2)
C: Serial Port  1 (COM1)
D: Serial Port  2 (COM2)
```

Epson EX-800

- "Setup", "Print", "Output Unit", "Printer", "Epson EX-800"

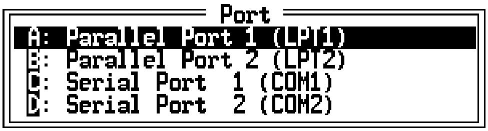
You can select here between:

```
Port
A: Parallel Port 1 (LPT1)
B: Parallel Port 2 (LPT2)
C: Serial Port  1 (COM1)
D: Serial Port  2 (COM2)
```

Epson FX-80

- "Setup", "Print", "Output Unit", "Printer", "Epson FX-80"

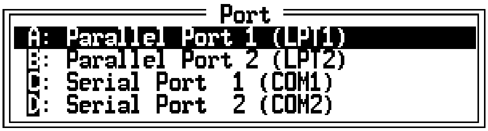
You can select here between:



NEC LP60/P70

- "Setup", "Print", "Output Unit", "Printer", "NEC P60/P70"

The printers PRT 294 and PRT 295 can be initialized with this setting. You can select here between:



Lines/Page

- ”Setup”, ”Print”, ”Lines/Page”

You can define the number of lines per page corresponding to the paper format in your printer here using the line editor.

Step 1 <Return> (Start edit)

Step 2 Enter number of lines (0-99)

Step 3 Terminate with <Return>

Recommendation: Z-folded DINA4 paper, up to 68 lines/page
for DRU 1200, up to 85 lines/page

Note No more than 32 characters of signal comment may be entered when using DIN A4 printers as otherwise a pagination after 132 characters causes a line displacement.
The AEG printers DRU 120 and DRU 292 are affected by this.

Form Feed

- ”Setup”, ”Print”, ”Form feed”

Depending on the printer type used, output can be with or without form feed. If form feed is switched off, the print program outputs empty lines to correspond to the number of lines.

You can toggle between ”on” and ”off”.

4.8.5 Colors

- "Setup", "Colors"

If you have a color monitor with the corresponding module at your PADT, you can set the colors yourself. You can set colors for the pulldown windows, help windows and message windows to suit yourself.

The extension "/COL" is required in the software call.

Furthermore, you can also set the monitor parameter "/GR" for shades of grey and "/BW" for black-and-white in the call.
The system uses fixed settings in calls with these parameters.

Recommendation: Color monitor "/COL", Liquid crystal display "/GR", Monochrome display "/BW".

If the software is called with the parameter /COL, a further pulldown menu appears for each type of window. The following parts of a window can be selected for the color setting here:

Pulldown window

```
Window Part
Frame
Normal Text
Inverted Text
Selection Letter
INverse Selection Letter
Background
InVerted Background
```

Help window

```
Window Part
Frame
Normal Text
Inverted Text
Catchword
Background
InVerted Background
```

Message window

```
Window Part
Frame
Normal Text
Inverted Text
Background
InVerted Background
```

When one of these menu lines is selected, a window with a choice of colors appears. You can select one of these colors.
The current setting is displayed in the demonstration window. You leave the ”color” window with <Esc>.
The new color settings appear after leaving the Setup pulldown menu.

Chapter 5
For Users of BSW < 6.0

5.1 Loadable Basic Software Version 6.0

The loadable basic software can only be used with ALU 021 and ALU 071. These central processing units can only be used in the A500 programmable controllers. The ALU 821, ALU 150, ALU 286, ALU 011 and ALU 061 central processing units can only be used with basic software versions less than 6.0.

5.2 Configure Basic Software

Module configuration is not possible using the software for basic software versions less than 6.0. It is only possible with ALU 021 or ALU 071 and thus with BSW 6.0.

5.3 Dolog SFB Address List

The function ”Edit”, ”Dolog SFB Address List” is only available for basic software versions less than 6.0.

Note If an older AKF version of the files ’DOLOG1/2.bau’ exists in the station, these files must be deleted.
If you use your own SFBs in connection with a BSW V5.x, you must enter these SFBs in the Dolog SFB address list again and save the address table. A new file ’DOLOG1/2.bau’ occurs.

5.4 POS 102/POS 112 with AKF35 and BSW Version 6.0

Positioning is only possible for the POS 102 and POS 112 modules with ALU 021 and ALU 071 (loadable basic software, version 6.0) and therefore can only be used for A500.

5.5 Intelligent Function Modules (Front)

The procedure described below should be used for intelligent function modules with a basic software version less than 6.0 and with AKF35 version 6.0. It is also valid for the intelligent function modules to be used with BSW 5.0 in front connection technique.

5.5.1 Flowchart

The following flowchart roughly describes the work flow when using intelligent function modules (front connection technique) together with a basic software version < 6.0. Step-by-step processing is documented in chapter 5.5.2 ff.

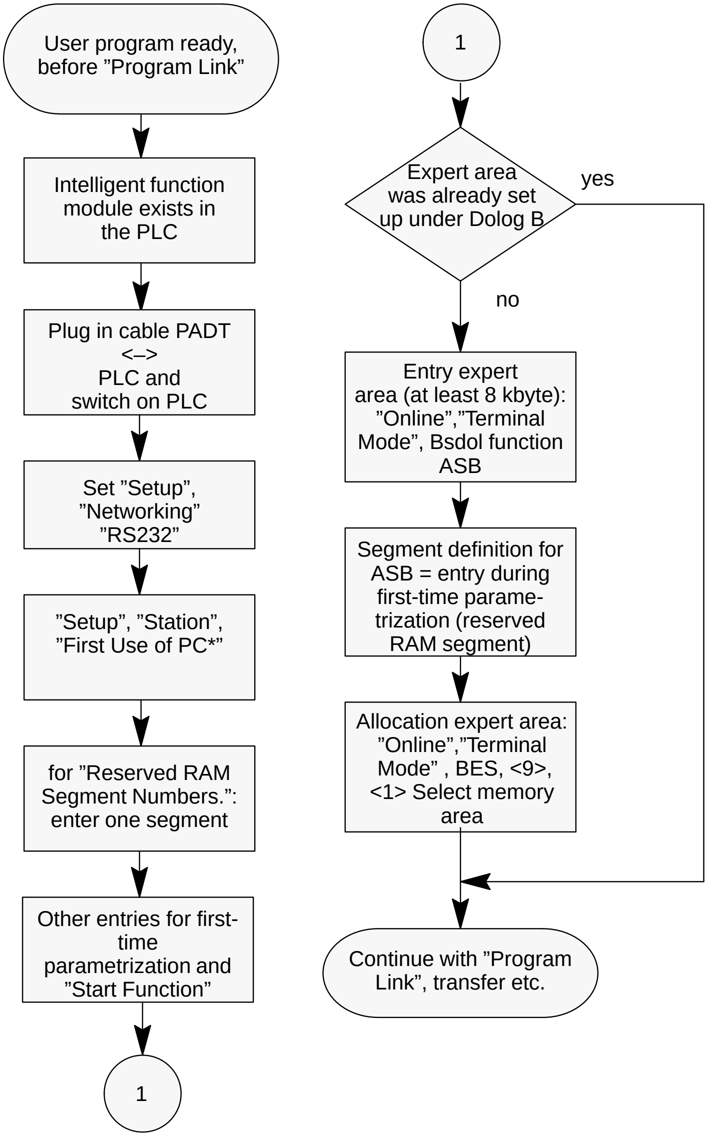


Figure 20 Special Characteristics BSW < 6.0 with intelligent function modules

5.5.2 First-time Parametrization of PLC

The first-time PLC parametrization must occur at the latest when the user progam is finished. The areas which are programmed with Dolog B or which are to be processed with the Terminal Mode must be taken into consideration.

With the Terminal Mode you set up a memory area for the intelligent function modules of the front connection technique. This memory area must be withheld from the AKF part. This is done by entering at least one segment in ”Reserved Segment Numbers.” during the first-time PLC parametrization (”Setup”, ”Station”, ”First Use of PC*”).

```
                         First Use of PC*
Start Function
RAM/EPROM Version          : RAM
EPROM-Segment numbers      :
RAM-Segment numbers        : 5,6
Set up RAMZU-PADT          : Yes
Set up RAMZU-SEAB          : Yes
Reserved Segment numbers   : 7,8,13,14
```

☞ **Note** Make sure that you note the numbers of the reserved segments.

After ending the first-time parametrization, the AKF memory assignment is displayed in a window, e.g.:

```
              PC* occupancy for : 271598/05
Memory area   Segment                      Usage
1-2            5-6                          AKF/RAM
30             15                           RAMZU-SEAB
31             15                           RAMZU-PADT
32             16                           SYKON
```

5.5.3 Set Up ”RAMZU Expert” with Terminal Mode

A step-by-step description of how to make your entries in the function ”Terminal Mode” follows.

5.5.3.1 Select Terminal Mode

Select the function ”Online”, ”Terminal Mode” if your progrmmable controller is connected. The PC now responds online with ”Dolog B:”.

5.5.3.2 Select Memory Area

An unused memory area must first be found.

Step 1 Enter DSB for ”Document Memory Area”.

React. A printout of all unused and reserved memory areas and their size is displayed (see example below).

```
DOLOG B:DSB

FREE   RAM    AREA:

   1:  12768 TO   12768
   7:      1 TO   32768
   8:      1 TO   32768
  13:      1 TO   32768
  14:      1 TO   32768
  15:      1 TO   18432

FREE   EPROM    AREA:

NUMBER OF DIGIT. VALUES:  10000

RESERVED     STORAGE     AREA:

STORAGE     AREA  1 (RAM)          5:      1 TO   32768
STORAGE     AREA  2 (RAM)          6:      1 TO   32768
STORAGE     AREA 30 (RAM)         15:  18433 TO   22528
STORAGE     AREA 31 (RAM)         15:  22529 TO   32768
STORAGE     AREA 32 (RAM)         16:      1 TO   32768

DOLOG B:
```

Unused segments
(Entered in AKF during first-time parametrization under ”Reserved Segment Numbers”)

Used memory areas
(assigned by AKF)

5.5.3.3 Set Up Memory Area

A memory area is now assigned in the RAM area for the intelligent function modules. A whole segment (32,768 bytes) is assigned to the memory area with the function ASB.

Step 1 Enter ASB for ”Set Up Memory Area”.

React. You are asked for the number of the memory area to be set up.

Step 2 Enter a memory area number. The assigned memory areas (from 1 ... 32) are displayed in the function DSB under ”Reserved Memory Areas”. (In the example 5 was selected)

React. You are asked for the number and size of the segment assigned to the memory area.

Step 3 Enter one of the segments which you find in the function DSB under ”Free RAM Areas”. Note that the defined segment has 32 768 unused bytes. (In the example 7 was selected)

React. You are asked for the starting address of the memory area.

Step 4 Enter at `from` `1` and at `to` `32768`.

React. The next memory area is requested.

Step 5 Enter `E` for End.

```
DOLOG B:ASB

NUMBER   OF   STORAGE    AREA: 5
STORAGE     AREA  5                         NO     ENTRY  SEGMENT: 7
FRM: 1
TO : 32768
TYPE=RAM
STORAGE     AREA  6                         NO     ENTRY  SEGMENT: E

DOLOG B:
```

5.5.3.4 Enter ”RAMZU Expert”

The memory area just defined must now be assigned to the intelligent function modules. This is done with the function BES.

Step 1 Enter BES for ”Equipment List”.
The following menu appears:

```
  DOLOG B:BES
*** I/O-Occupancy list,          ***
1: Delete list area from ... to          2: Display list area from ... to
3: Display or change list elements       4: Delete error messages
5: Error diagnosis                       6: Initialize
7: Trace function                        8: Special functions
9: Expert functions
H: Help function                         E: Return to DOLOG
Input:
```

Step 2 Enter 9 for ”Expert Functions”.
The next menu appears.

```
Input: 9

  1:   RAMZU experts
  2:   Expert state
  3:   DPM access
  E:   End

  Input: 1
```

☞ **Note** The equipment list must be initialized if these functions are to be executed. If the request for ”input” appears again here, first enter 6 for ”Initialize Entries”.

Step 3 Enter 1 for ”RAMZU Experts”.

React. All ”Used RAM Areas” are displayed.

```
SA  1 =  5 :     1 - 32768            SA  2 =  6 :     1 - 32768
SA  5 =  7 :     1 - 32768            SA 30 = 15 : 18433 - 22528
SA 31 = 15 : 22529 - 32768            SA 32 = 16 :     1 - 32768

No storage assign.

End : E      SA list : N     Clear   : -
SA minimum length 2000
Up to 8 storage areas separated by blanks

Input :
```

Step 4 Enter the memory area which you set up with ASB (in example 5).

Step 5 Enter E for ”End” three times at ”Select”.

```
  Storage areas for experts     : 5

  End : E      SA list : N     Clear   : -
  SA minimum length 2000
  Up to 8 storage areas separated by blanks

  Input : E

  1:   RAMZU experts
  2:   Expert state
  3:   DPM access
  E:   End

  Input: E

Input: E
Initialising in progress....
Initialising completed without errors.

  DOLOG B:
```

Step 6 Leave the Terminal Mode with the <F9> key.

The programmable controller is now set up for the intelligent function modules. You can now transfer your user program to the PLC.

5.6 Connection User Program Dolog B/ Dolog AKF

Note This function can be used without limitations with basic software versions < 6.0.
The loadable basic software may no longer be changed after Dolog B program creation for basic software versions ≥ 6.0. Dolog B must be programmed online after loading the basic software.

It could be advisable, for example, to use Dolog AKF program parts in addition to Dolog B Vlists for system extensions. The following rules should be followed in this case.

- Separate memory areas are used in the PLC for Dolog AKF and Dolog B.
- The Dolog AKF program and the Dolog B Vlist may reside in any segments. Note that the number of the AKF memory area is always located in the load blocks (see example) at ”R2”.
- The program is started in the Dolog B memory area and continued to its end in the Dolog AKF memory area.
- The segments reserved for Dolog B in ”Reserved Segment Numbers.” must be entered in the Dolog AKF first-time parametrization.
- The Dolog B Vlist is started with the Bsdol function ”S” in the memory area. Only the memory areas contained in the Vlist but not the AKF memory areas may be specified for ”S”.

An example with an explanation of the function follows. The program structure of the Dolog Vlsit can be seen in the example.

Example:

☞ **Note** This function can be used without limitations for basic software versions < 6.0.
The loadable basic software may not be changed after the Dolog B program creation for basic software versions ≥ 6.0. Dolog B must be programmed online after loading the basic software.

Memory Area 1 **Memory Area 10**

Memory Area 1	Memory Area 10
Dolog AKF program	Dolog B program

```
Dolog B
program
:
:
LA 1
EF 11
R1 X
R2 1
:
LA1
EF 11
R1 Y
R2 1
:
SWB
EF 11
WB X
WA Y
:
SZW
WB X
WA Y
END
```

Both SFB LA1 take care of the correct jump address.

You enter the Memory Area No. of the Dolog AKF program at R1. After starting the user program with Bsdol function ”S” the created address is redefined.
For that the constant at R2 (in example ”1”) is written as jump address in RX or RY.

Marker 11 = constant ”1”

Jump to AKF program
”SWB = Far Jump, needs for terminating the SFB ”SZW”. SWB allows to jump over program parts. The target may be in another memory area. The function is executed with EF = 1.

A ”SWB” always needs a ”SZW”. Otherwise there you get an error message during program start.

The END-block is not executed. It only serves as ”optical program end”

Part IV
Formal Operands of SFBs

Chapter 1
SFB Formal Operands

This chapter contains the formal operands for all the standard function blocks (SFBs) available in version 6.x. The SFBs are in alphabetical order.

1.1 General Information

The basic structure of the tables is shown using INV as an example:

INV Invert Word (SFB101)

Formal operand	**Identi- fier**	**Meaning**
INV		Operation (call)
EF	Bit addr.	Enable
DW	MW addr.	<DW> = input value
DK	MW addr.	<DK> = <DW> inverted

Detailed documentation can be found in the following manuals:

A350/A500
Dolog AKF Standard Function Blocks
(AKF35EN Version 5.x)
Block Library
A91M.12-271891

[8]A500
Dolog AKF Standard-Funktionsbausteine
(AKF35 Version 6.x)
Bausteinbibliothek
A91M.12-279346

Note The block library A91M.12-279346 contains the new standard function blocks of version 6. If you want to take advantage of all the features of AKF35 version 6.0, you need both manuals.

8) in german language

1.2 Tables of the Formal Operands

A5DBS Data Transfer A500 → DBS (SFB50)

Formal operand	Identifier	Meaning
A5DBS		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Hardware reset on DBS 001
EASY	Bit addr.	Transmission asynchronously to DBS program
SP	MW addr.	Slot reference of DBS (higher address)
WTUE	MW addr.	Monitoring time for the PEAB (x * 100 msec)
QWA	MW addr.	Source 1st word address in DBS
LW	MW addr.	Length of the word field
ZWA	MW addr.	Target 1st word address in PLC
QBA	MW addr.	Source 1st bit address in DBS
LB	MW addr.	Length of the bit field
ZBA	MW addr.	Target 1st bit address in PLC
VI7	MW addr.	Internal organization information
RDY	Bit addr.	<RDY> = 0: Data transmission running <RDY> = 1: Output buffer empty or transmission aborted,
MRYA	Bit addr.	Synchronization bit for all blocks which access the same hardware. Ensure correct wiring.
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

AB1 Rate of Change Limit 1st Order (SFB350)

Formal operand	Identifier		Meaning
AB1			Operation (call)
STAT	Word addr.		Status control loop: -2 = Halt, -1 = Reset, 0 = Initial, 1 = Running
RST	Bit addr.		Operating mode Reset (”1” = Reset)
HALT	Bit addr.		Operating mode Halt (”1” = Halt)
PARA	PAB?		Data structure parameter (see below)
X	FWord addr.		Input
YRST	FWord addr.		Reset value output
Y	FWord addr.		Output
YAO	Bit addr.		Message: Output, reached upper limit = ”1”
YAU	Bit addr.		Message: Output, reached lower limit = ”1”
AF	Bit addr.		AF = ”1”: error
WAF	Word addr.		Error code
VI	VIA?	16 byte	Data structure organization information

PAB

Element	Element type	Symbol suggestion	Meaning
PABn			Data structure parameters of AB1, n = 50
PABn.1	Floatword	GVM	Maximum rate of limit change (maximum x’)
PABn.2	Floatword	OG	Upper limit
PABn.3	Floatword	UG	Lower limit

AB2 Rate of Change Limit 2nd Order (SFB351)

Formal operand	Identi-fier		Meaning
AB2			Operation (call)
STAT	Word addr.		Status control loop: -2 = Halt, -1 = Reset, 0 = Initial, 1 = Running
RST	Bit addr.		Operating mode Reset (”1” = Reset)
HALT	Bit addr.		Operating mode Halt (”1” = Halt)
PARA	PABB?		Data structure parameters (see below)
X	FWord addr.		Input
YRST	FWord addr.		Reset value output
Y	FWord addr.		Output
AF	Bit addr.		AF = ”1”: error
WAF	Word addr.		Error code
VI	VIA?	16 bytes	Data structure organization information

PABB

Element	Element type	Symbol suggestion	Meaning
PABBn			Data structure parameters of AB2, N = 50
PABBn.1	Floatword	GVM	Maximum rate of limit change (maximum x’)
PABBn.2	Floatword	GBM	Maximum acceleration of limit change (maximum x’’)

ABS Absolute Value Generation Word (SFB152)

Formal operand	Identi-fier	Meaning
ABS		Operation (call)
WE	MW addr.	<WE> = signed number
WA	MW addr.	<WA> = absolute value of <WE>
AV	Bit addr.	AV = 0: <WE> is a positive number AV = 1: <WE> is a negative number

ACOS Arc Cosine Function (SFB280)

Formal operand	Identi-fier	Meaning
ACOS/ARCCOS		Operation (call)
EF	Bit addr.	Enable
GE	MF addr.	Input quantity
GA	MF addr.	Output quantity in radian measure
AF	Bit addr.	AF=1: error
WAF	MW addr.	Error code

ADE Addition Word (SFB156)

Formal operand	Identi-fier	Meaning
ADE		Operation (call)
WE2	MW addr.	<WE2> = summand 2
WE1	MW addr.	<WE1> = summand 1
WA	MW addr.	<WA> = result
AF	Bit addr.	AF = 1: error

AEK Change Signal (SFB253)

Formal operand	Identi-fier	Meaning
AEK		Operation (call)
EF	Bit addr.	Enable
EQ	Bit addr.	Acknowledgement
BK	Bit addr.	Smallest bit address of bit string (corresponds to most significant bit)
BL	Bit addr.	Last bit address of bit string (corresponds to least significant bit)
WZ	MW addr.	<WZ> = previous state of bit string BK ... BL
WG	MW addr.	<WG> = change signal 1 $\rightarrow$ 0
WK	MW addr.	<WK> = change signal 0 $\rightarrow$ 1
AE	Bit addr.	AE = 1 if change (1 for a scan)

AEM Change Signal 16 Bits (SFB115)

Formal operand	Identi-fier	Meaning
AEM		Operation (call)
EF	Bit addr.	Enable
EQ	Bit addr.	Acknowledgement
BK	Bit addr.	Smallest bit address of bit string (corresponds to most significant bit)
BL	Bit addr.	Last bit address of bit string (corresponds to least significant bit)
WZ	MW addr.	<WZ> = previous state of bit string BK ... BL
AE	Bit addr.	AE = 1 if change (1 for a scan)

AEQ Equivalence Word (SFB108)

Formal operand	Identi-fier	Meaning
AEQ		Operation (call)
WI	MW addr.	Actual value
WV	MW addr.	Comparison value
AK	Bit addr.	Output <WI> < <WV>
AA	Bit addr.	Output <WI> = <WV>
AG	Bit addr.	Output <WI> > <WV>

ALARM Entry in the Alarm List B500 (SFB260)

Formal operand	Identi-fier	Number	Meaning		
ALARM			Operation (call)		
EF	Bit addr.		Enable		
PVN	MW addr.		<PVN>	=	PV-number
EO	Bit addr.		EO	=	1: upper limit violation
OG	MW addr.	1 or 2	<OG>	=	upper limit, for MF format reserve 2 MW
EU	Bit addr.		EU	=	1: lower limit violation (trigger input)
UG	MW addr.	1 or 2	<UG>	=	lower limit, for MF format reserve 2 MW
VI1	MW addr.		<VI1>	=	previous states of signals EO, EU

ASIN Arc Sine Function (SFB279)

Formal operand	Identi-fier	Meaning
ASIN/ARCSIN		Operation (call)
EF	Bit addr.	Enable
GE	MF addr.	Input quantity
GA	MF addr.	Output quantity in radian measure
AF	Bit addr.	AF=1: error
WAF	MW addr.	Error code

ATAN Arc Tangent Function (SFB281)

Formal operand	Identi-fier	Meaning
ATAN/ARCTAN		Operation (call)
EF	Bit addr.	Enable
GEZ	MF addr.	Input quantity (counter dY)
GEN	MF addr.	Input quantity (denominator dX)
GA	MF addr.	Output quantity in radian measure
AF	Bit addr.	AF=1: error
WAF	MW addr.	Error code

AUS Direct Output to a Pin String (SFB188)

Formal operand	Identi-fier	Meaning
AUS		Operation (call)
EF	Bit addr.	Enable
AK	Bit addr.	Starting address of bit string (outputs)

AUTO_POS Automatic Operation POS 102/112; A500 as of BSW 6.0 (SFB64)

Formal-operand	Identi-fier	Number	Meaning
AUTO_POS			Operation (call)
ER	Bit addr.		Reset (see configuration)
AKEN	Word addr.		Job code (see configuration)
TN / SP	TN addr.		for A500: <SP> = physical slot reference, 2 ... 160
ACHS	Word addr.		<ACHS> = axis number ”0” = common axis mode ”1” = single axis mode, axis 1 ”2” = single axis mode, axis 2
ESAT	Bit addr.		”0” = operating mode automatic ”1” = operating mode automatic-single step (see configuration)
UNTB	Bit addr.		”1” = interrupt traversing program
ANW	Bit addr.		Control selection of a traversing program
SS	Bit addr.		Start/stop flag (see configuration)
PARA	APOA?		Data structure ”parameters for SFB AUTO_POS” (see below)
RDY	Bit addr.		”1” = signal job executed; during the data transmission between PLC and POS, <RDY> = 0 (see configuration)
ACCE	Bit addr.		Acknowledgement, ”1” = job accepted (see configuration)
MELD	MPOA?		Data structure ”Signals from SFB AUTO_POS” (see below)
AUFA	Bit addr.		”1” = job error (see configuration)
VI	VIPO?		Data structure ”internal organization information”
AF	Bit addr.		Error marker, AF = ”1”: error
WAF	Word addr.		Error code, <WAF> = error number

APOA

Element	Element type	Symbol suggestion	Meaning
APOAn			Data structure parameters for SFB AUTO_POS n = 1, ..., 64
APOAn.1	Word	NR_VP	<> = Number of traversing program to be selected, 1 ... 65
APOAn.2	Word	NR_SATZ	<> = Record number where traversing program starts, see configuration (0 means: start with 1st record of traversing program, number > 0: start with record whose number is displayed)
APOAn.3	Word	OVERR_A	<> = override factor for automatic operation in % (0 ... 100)
APOAn.4	Bit	SATZUN	”1” = switch on record suppression, ”0” = switch off record suppression
APOAn.5 ... APOAn.13			Reserved

MPOA

Element	Element type	Symbol suggestion	Meaning
MPOAn			Data structure signals from SFB AUTO_POS n = 1, ..., 64
MPOAn.1	Word	H_BTR_VP	<> = Main operating mode for automatic, ”0” = common axis operation ”1” = single axis operation
MPOAn.2	Bit	VP_AN_HL	”1” = traversing program running, ”0” = traversing program halted or not started
MPOAn.3	Bit	BEW_AUTO	”1” = automatic operating mode active
MPOAn.4	Bit	BEW_AUEZ	”1” = automatic-single step operating mode active
MPOAn.5	Bit	SATZEND	”1” = wait for start command, e.g. after expiration of a record in automatic single-step operating mode or after selecting a program in automatic operating mode
MPOAn.6	Bit	M_30	”1” = traversing program terminated with M30
MPOAn.7 ... MPOAn.17			Reserved

AWA1 Analog Value Output to the MWA 16 PN -8 Bits- (SFB191)

Formal operand	Identi-fier	Meaning
AWA1		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Reset
SP	MW addr.	<SP> = slot reference of the MWA 16 PN, 2 ... 160
KA	MW addr.	<KA> = first channel to be output, 1 ... 16
KN	MW addr.	<KN> = number of channels to be output, 1 ... 16-<KA>
WE	MW addr.	<WE> = word for the 1st measured value (channel)
AF	Bit addr.	AF = 1: error

AWA3 Analog Value Output to the MWA 16 PN -10 Bits- (SFB192)

Formal operand	Identi-fier	Meaning
AWA3		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Reset
SP	MW addr.	<SP> = slot reference of the MWA 16 PN, 2 ... 160
KA	MW addr.	<KA> = first channel to be output, 1 ... 16
KN	MW addr.	<KN> = number of channels to be output, 1 ... 16-<KA>
WE	MW addr.	<WE> = word for the 1st measured value (channel)
AF	Bit addr.	AF = 1: error

AWA8 Analog Value Output to the DAU 104/108 (SFB197)

Formal operand	Identi-fier	Number	Meaning	
AWA8			Operation (call)	
EF	Bit addr		Block enable	
ER	Bit addr.		Reset	
SP	MW addr.		<SP> =	Slot; 1 or 2 ... 160 depending on ALU type used, corresponding to entry in the equipment list
KA	MW addr.		<KA> =	first channel to be output (channel 1 ... 4 or 1 ... 8)
KN	MW addr.		<KN> =	number of channels to be output (number 1 ... 4 or 1 ... 8)
VI3	MW addr.	3	Organization information; do not change contents!	
WA	MW addr.		<WA> = 1st value to be output	
SKA	MW addr.		<SKA> = Start of scale	
SKE	MW addr.		<SKE> = End of scale	
AF	Bit addr.		Error marker, AF = 1 if error	
WAF	MW addr.		<WAF> = error code	

AWE1 Analog Value Input from the ADU S9 (SFB190)

Formal operand	Identi-fier	Meaning		
AWE1		Operation <call>		
EF	Bit addr.	Enable		
ER	Bit addr.	Reset		
SP	MW addr.	<SP>	=	Slot reference of ADU S9, 2 160
KA	MW addr.	<KA>	=	First channel to be read, 2....256
KN	MW addr.	<KN>	=	Number of channels to be read, 2...256-<KA>*)
WA	MW addr.	<WA>	=	Word for 1st measured value (channel)
AF	Bit addr.	AF = 1: error		

*) 1...16 for ADU S9 without module MWE 32
2...256 for ADU S9 with module MWE 32

AWE13 Analog Value Input with ADU I13.2 (SFB259)

Formal operand	Identi-fier	Number	Meaning
AWE13			Operation <call>
EF	Bit addr.		Enable
ER	Bit addr.		Reset
BT	Bit addr.		Commission: start conversion for 0/1-edge
SP	MW addr.		Slot reference of ADU I13.2
KA	MW addr.		<KA> = 1st channel to be input, 1 ... 256*)
KN	MW addr.		<KN> = number of channels to be input, 1 ... 255-<KA>*)
PAR	MW addr.		<PAR> = parameter for gain and characteristics of measuring point
MRY	Bit addr.		Memory signal: 0: a AWE13 block converting; 1: no AWE13 block converting
VI5	MW addr.	5	Internal organization information
WA	MW addr.		Word for first measured value (channel). One word must be reserved for each converting channel for the result of the conversion.
RDY	Bit addr.		Ready signal: 0: this block converting; 1: not converting
AF	Bit addr.		AF = 1: error
WAF	MW addr.		Error code

*) corresponding to number of equipped channels

AWE16 Analog Value Input from the ADU 115/116, DAU 104 (SFB196)

Formal operand	Identi-fier	Number	Meaning
AWE16			Operation (call)
EF	Bit addr.		Block enable
ER	Bit addr.		Reset
EW	Bit addr.		Measuring range setting - EW = 0: without suppressed zero point (without open-circuit monitoring) - EW = 1: with suppressed zero point (with open-circuit monitoring)
EP	Bit addr.		- EP = 0: measuring range bipolar - EP = 1: measuring range unipolar
SP	MW - addr.		<SP> = slot location; 1 or 2 ... 160, depending on ALU used; corresponds to entry in equipment list
KA	MW addr.		<KA> = first channel to be read (channel 1 ... 16 or 8 for DAU 104)
KN	MW addr.		<KN> = number of channels to be read (number 1 ... 16 or 8 for DAU 104)
VI3	MW addr.	3	Organization information; contents may not be changed
SKA	MW addr.		<SKA> = start of scale
SKE	MW addr.		<SKE> = end of scale
OG	MW addr.		<OG> = upper limit
UG	MW addr.		<UG> = lower limit
WE	MW addr.		<WE> = 1st converted measured value (address of 1st channel)
AO	Bit addr.		AO = 1: upper limit violated
AU	Bit addr.		AU = 1: lower limit violated
AF	Bit addr.		Error marker, AF = 1 if error
WAF	MW addr.		<WAF> = error status

AWE4 — Analog Value Input with AEM 2511, EMU 2610 (SFB258)

Formal operand	Identi-fier	Meaning
AWE4		Operation <call>
EF	Bit addr.	Enable
ER	Bit addr.	Reset
EP	Bit addr.	Switch operating mode 0: bipolar/1: unipolar
EW	Bit addr.	Measuring range setting for unipolar operating mode
EV	Bit addr.	Switchover gain V = 1/ V = 8
SP	MW addr.	Slot reference of AEM 2511, 2 ... 160
KA	MW addr.	<KA> = 1st channel to be input, 1 ... 256*)
KN	MW addr.	<KN> = number of channels to be input, 1 ... 255-<KA>*)
WA	MW addr.	Word for the first measured value (channel).
AD	Bit addr.	AD = 1: open-circuit
AF	Bit addr.	AF = 1: error

*) corresponding to number of equipped channels

BALK — Bar Graph of a Marker Floating Point Word (SFB220)

Formal operand	Identi-fier	Meaning
BALK		Operation (call)
GE	MF addr.	<GE> = MF value to be displayed as bar
GSKA	MF addr.	<GSKA> = Start of scale (starting value of bar)
GSKE	MF addr.	<GSKE> = End of scale (end value of bar)
ZABL	MW addr.	<ZABL> = Number of chars. for the bar (bar length): max. 100 chars.; do not specify a value in line 1 of the output file.
SBN	MW addr.	<SBN> = Memory range number of the bar file
DNR	MW addr.	<DNR> = File number of bar file
ZN1	MW addr.	<ZN1> = 1st line number for special control chars. in the bar file; 2 lines must be specified.
ZN2	MW addr.	<ZN2> = 1st line number for general control chars. in the bar file; 4 lines must be specified.
PI7B	MW addr.	<PI7B> = 1st word internal organization information; is specified in the signal file as P-parameter

BAUS Direct Output to a Pin String, Modnet 1/SFB (SFB194)

Formal operand	Identi-fier	Meaning
BAUS		Operation (call)
EF	Bit addr.	Enable
AK	Bit addr.	Starting address of bit string (I/O)

BAW Bit Output from a Word (SFB133)

Formal operand	Identi-fier	Meaning
BAW		Operation (call)
EF	Bit addr.	Enable
DE	MW addr.	The bit is read from this M word
DZ	MW addr.	<DZ> = Bit number
AS	Bit addr.	Output bit

BEIN Direct Input a Pin String, Modnet 1/SFB (SFB193)

Formal operand	Identi-fier	Meaning
BEIN		Operation (call)
EF	Bit addr.	Enable
EK	Bit addr.	Starting address of bit string (I/O)

BEW Bit Input to Word (SFB134)

Formal operand	Identi-fier	Meaning
BEW		Operation (call)
EF	Bit addr.	Enable
ES	Bit addr.	Input bit
DZ	MW addr.	<DZ> = Bit number
DE	MW addr.	The bit is written to this M word

BISAx Bit Collector Bit (SFB164-166)

Formal operand			Identi-fier	Meaning
BISA4	BISA8	BIS16		Operation (call)
EF	EF	EF	Bit addr.	Enable
E1	E1	E1	Bit addr.	1st input bit
E2	E2	E2	Bit addr.	2nd input bit
E3	...	...	...	
E4	E8	E16	Bit addr.	Xth input bit
A	A	A	Bit addr.	Target bit for E1

BIVEx Bit Distributor Bit (SFB167-169)

Formal operand			Identi-fier	Meaning
BIVE4	BIVE8	BIV16		Operation (call)
EF	EF	EF	Bit addr.	Enable
E	E	E	Bit addr.	1st bit of source bit string
A1	A1	A1	Bit addr.	Target bit for E
A2	A2	A2	Bit addr.	Target bit for E+1
...	...	...	...	
A4	A8	A16	Bit addr.	Target bit for E+(x-1)

BSPC1 Data Transmission Backup Controller → Viewstar 100 (SFB54)

Formal operand	**Identi-fier**	**Meaning**
BSPC1		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Hardware reset on BUR 001
SP	MW addr.	Slot reference
KA	MW addr.	Control channel
ELF	Bit addr.	Control device enable
TUE	MF addr.	Monitoring time for PEAB
WSPC	MF addr.	Setpoint value of PLC
KZ	MF addr.	Gain disturbance
AZ	MF addr.	Operating point disturbance
UZ	MF addr.	Neutral zone W-X
XSKA	MF addr.	Start of scale; presetting: 0; 77 = switched off
XSKE	MF addr.	End of scale; presetting: 100; 77 = switched off
EBU1	Bit addr.	EBU1/2: 00 = follow-up control; else SPC control
EBU2	Bit addr.	with coded backup operating mode: 10 = HAND,
		01 = AUTO, 11 = KASK
EGUM	Bit addr.	Changeover to limit monitoring
WKB	MW addr.	Loop display number Viewstar 100
VI1	MW addr.	Internal organization information
FERN	MW addr.	Remote control output for Viewstar B500
AY0	Bit addr.	Controller inhibit of BUR
ABS	Bit addr.	BUR or transmission disturbance
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

BSPC5 Data Transmission Backup Controller → Viewstar B500 (SFB55)

Formal operand	**Identi-fier**	**Meaning**
BSPC5		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Hardware reset on BUR 001
SP	MW addr.	Slot reference
KA	MW addr.	Control channel
ELF	Bit addr.	Control device enable
TUE	MF addr.	Monitoring time for PEAB
WSPC	MF addr.	Setpoint value of PLC
KZ	MF addr.	Gain disturbance
AZ	MF addr.	Operating point disturbance
UZ	MF addr.	Neutral zone W-X
XSKA	MF addr.	Start of scale; presetting: 0; 77 = switched off
XSKE	MF addr.	End of scale; presetting: 100; 77 = switched off
EBU1	Bit addr.	EBU1/2: 00 = follow-up control; else SPC control
EBU2	Bit addr.	with coded backup operating mode: 10 = HAND, 01 = AUTO, 11 = KASK
EGUM	Bit addr.	Changeover limit monitoring
WH	MF addr.	Manual setpoint value*)
YH	MF addr.	Manual control output*)
XOG	MF addr.	Upper limit of X (actual value)
XUG	MF addr.	Lower limit of X (actual value)
H	Bit addr.	Manual operating mode Viewstar B500
A	Bit addr.	Auto operating mode Viewstar B500
K	Bit addr.	Cascade operating mode Viewstar B500
VI1	MW addr.	Internal organization information
FERN	MW addr.	Remote control output for Viewstar B500
AY0	Bit addr.	Controller inhibit of BUR
ABS	Bit addr.	BUR or transmission disturbnace
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code
W	MF addr.	Output effective setpoint value
X	MF addr.	Output actual value
Y	MF addr.	Output control output
KP	MF addr.	Gain*)
TN	MF addr.	Reset time*)
TV	MF addr.	Derivative action time*)
T1	MF addr.	Delay time D-part*)
AXOG	Bit addr.	Upper limit monitoring X
AXUG	Bit addr.	Lower limit monitoring X
SH	Bit addr.	Manual status for color change
SA	Bit addr.	Auto status for color change
SK	Bit addr.	Cascade status for color change

*) can be changed with BUR or Viewstar B500/A500 depending on ELF

BWEIN Direct Input a Pin String to a Word, Modnet 1/SFB (SFB195)

Formal operand	Identi-fier	Meaning
BWEIN		Operation (call)
EF	Bit addr.	Enable
EK	Bit addr.	Starting address of the bit string (I/O)
WA	MW addr.	<WA> = Binary values of the 16 signals

COS Cosine (SFB277)

Formal operand	Identi-fier	Meaning
COS		Operation (call)
EF	Bit addr.	Enable
GE	MF addr.	Input quantity in radian measure
GA	MF addr.	Output quantity
AF	Bit addr.	AF=1: error
WAF	MW addr.	Error code

DABS Absolute Value Generation of Double Word (SFB153)

Formal operand	Identi-fier	Meaning
DABS		Operation (call)
WE	MD addr.	<WE> = Signed number
WA	MD addr.	<WA> = Amount of <WE>
AV	Bit addr.	AV = 0: <WE> is a positive number AV = 1: <WE> is a negative number

DADD Addition Double Word (SFB160)

Formal operand	Identi-fier	Meaning
DADD		Operation (call)
DE1	MD addr.	<DE1> = M double word summand 1
DE2	MD addr.	<DE2> = M double word summand 2
DA	MD addr.	<DA> = M double word result
FM	Bit addr.	FM = 1: error

DAEQ Equivalence Double Word (SFB109)

Formal operand	Identi-fier	Meaning
DAEQ		Operation (call)
WI	MD addr.	Actual value
WV	MD addr.	Comparison value
AK	Bit addr.	Output <WI> < <WV>
AA	Bit addr.	Output <WI> = <WV>
AG	Bit addr.	Output <WI> > <WV>

DBSA5 Data Transmission DBS → A500 (SFB51)

Formal operand	Identi-fier	Number	Meaning
DBSA5			Operation (call)
EF	Bit addr.		Enable
ER	Bit addr.		Hardware reset on DBS 001
EASY	Bit addr.		Transmission asynchronous to DBS program
SP	MW addr.		Slot reference of DBS
WTUE	MW addr.		Monitoring time for PEAB (x * 100 msec)
ZWA	MW addr.		Target 1st word address in PLC
LW	MW addr.		Length of word field
QWA	MW addr.		Source 1st word address in DBS
ZBA	MW addr.		Target 1st bit address in PLC
LB	MW addr.		Length of bit field
QBA	MW addr.		Source 1st bit address in DBS
VI7	MW addr.	7	Internal organization information
RDY	Bit addr.		<RDY> = 0: Data transmission running; <RDY> = 1 Output buffer empty or transmission aborted
MRYD	Bit addr.		Synchronization bit for all blocks which access the same hardware. Ensure correct wiring.
AF	Bit addr.		AF = 1: error
WAF	MW addr.		Error code

DCR Decrementer (SFB155)

Formal operand	**Identi-fier**	**Meaning**
DCR		Operation (call)
EF	Bit addr.	Enable
WE	MW addr.	Input M word
WA	MW addr.	Output M word

DDIV Division Double Word (SFB163)

Formal operand	**Identi-fier**	**Meaning**
DDIV		Operation (call)
DE1	MD addr.	<DE1> = M double word dividend
DE2	MD addr.	<DE2> = M double word divisor
DQ	MD addr.	<DA> = M double word quotient
DR	MD addr.	<DR> = Division remainder
FM	Bit addr.	FM = 1: error

DGW1 Convert Double Word to Floating Point Word (SFB229)

Formal operand	**Identi-fier**	**Meaning**
DGW1		Operation (call)
EF	Bit addr.	Enable
DWE	MD addr.	Input quantity double word 32 bits
GA	MF addr.	Floating point output quantity

DIE Division Word (SFB159)

Formal operand	**Identi-fier**	**Meaning**
DIE		Operation (call)
WE2	MW addr.	<WE2> = Divisor
WE1	MW addr.	<WE1> = Dividend
WA	MW addr.	<WA> = Quotient
WR	MW addr.	<WR> = Division remainder
AF	Bit addr.	AF = 1: error

DLAx Load Double Word, Effect see corresponding LAx (SFB141-144)

Formal operand	Identi-fier	Meaning
DLA1,		Operation (call)
EF	Bit addr.	Enable
R2	MW addr.	Source = constant
R1	MD addr.	Target see configuration
DLA2		Operation (call)
EF	Bit addr.	Enable
R2	MD addr.	Source see configuration
R1	MD addr.	Target see configuration
DLA4		Operation (call)
EF	Bit addr.	Enable
R2	MD addr.	Source see configuration
R1	MW addr.	Target see configuration
DLA5		Operation (call)
EF	Bit addr.	Enable
R2	MW addr.	Source see configuration
R1	MD addr.	Target see configuration

DLBW Load Bit String to Double Word (SFB132)

Formal operand	Identi-fier	Meaning
DLBW		Operation (call)
BK	Bit addr.	Smallest bit address of bit string (corresponds to most significant bit)
BL	Bit addr.	Last bit address of bit string (corresponds to least significant bit)
WA	MD addr.	Output M double word

DLWB Load Double Word to Bit String (SFB130)

Formal operand	Identi-fier	Meaning
DLWB		Operation (call)
WE	MD addr.	Input M double word
BK	Bit addr.	Smallest bit address of bit string (corresponds to most significant bit)
BL	Bit addr.	Last bit address of bit string (corresponds to least significant bit)

DMUL Multiplication Double Word (SFB162)

Formal operand	Identi-fier	Meaning
DMUL		Operation (call)
DE1	MD addr.	<DE1> = M double word multiplicant
DE2	MD addr.	<DE2> = M double word multiplicator
DA	MD addr.	<DA> = M double word result
FM	Bit addr.	FM = 1: error

DOZ1 Data Transfer DOZ ↔ A500 (SFB58)

Formal operand	**Identi-fier**	**Meaning**
DOZ1		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Hardware reset DOZ 001
SP	MW addr.	Slot reference
KA	MW addr.	Channel address
EZRD	Bit addr.	Reset actual value with 0/1-edge; 1-signal must be present at least 110 msec, irrelevant of the scan time
EQ	Bit addr.	Channel-dependent acknowledgement.; channel = <KA>
ELF	Bit addr.	Enable for control device; 0: PLC; 1: control device and PLC
EVF	Bit addr.	Enable output amplifier: EVF/VFn 01: switch off VOR and END, 11: enable VOR and END; VFn = hardware input; VFn dominating EVF, disconnection of END and VOR is thus possible in case of backup.
EBUA	Bit addr.	Behavior if backup; 0: AVOR/AEND switched off; 1: count
DOZ1		Continued
TUE	MF addr.	PEAB monitoring time in seconds
TGEB	MF addr.	Watchdog of sensor; 0.1 ... 25.5 sec; (00: no monitoring)
KF	MF addr.	Correction factor, e.g. for effects on temperature, humidity
UF	MF addr.	Value = counter contents / UF; UF = 3600 $*$ f_{max} / max measuring range $*$ KF *)
VOR	MF addr.	Prestop value (prestop); AVOR off if <VOR> $\geq$ actual value *)
END	MF addr.	Endstop value (stop); AEND off if <END> $\geq$ actual value *)
IST	MF addr.	Actual value of counter
DFH	MF addr.	Flow rate per hour
VI1	MW addr.	Internal organization information
AVF	Bit addr.	Output is set depending on binary output VFn
VOR	Bit addr.	Output is set depending on VOR, EVF and VFn
AEND	Bit addr.	Output is set depending on END, EVF and VFn
ALG	Bit addr.	Control device intervention
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

*) Can be changed with DOZ or PLC depending on ELF

DR Three-Position Controller (SFB315)

Formal operand	Identi-fier	Number	Meaning
DR			Operation (call)
STAT	Word addr.		Status control loop: -2 = Halt, -1 = Reset, 0 = Initial, 1 = Running
BT	BZR?		Data structure operating modes (see below)
PARA	PDR?		Data structure parameters (see below)
W	FWord addr.		Setpoint value input
X	FWord addr.		Actual value input
NG	FWord addr.		Standardization quantity
XRR	FWord addr.		Reset value of feedback (%)
YHP	Bit addr.		Manual value for YP
YHN	Bit addr.		Manual value for YN
YP	Bit addr.		Output positive manipulated variable
YN	Bit addr.		Output negative manipulated variable
AF	Bit addr.		AF = ”1”: error
WAF	Word addr.		Error code
VI	VIC?	48 bytes	Data structure organization information

BZR

Element	Element type	Symbol suggestion	Meaning
BZRn.x			Data structure operating modes of DR (= ZR), n = 100
BZRn.1	Bit	Reset	Control input for reset operating mode (”1” = Reset)
BZRn.2	Bit	Hand	Control input for manual operating mode (”1” = Hand)
BZRn.3	Bit	Halt	Control input for halt operating mode (”1” = Halt)
BZRn.4	Bit	PID-Par	Use of PID parameter record (Pid-Par=1) (else feedback parameter active)

PDR

Element	Element type	Symbol suggestion	Meaning	
PDRn.x			Data struture parameters of DR, n = 50	
PDRn.1	Float word	Kp	Proportional rate (gain)	
PDRn.2	Float word	Tn	Reset time (in sec)	PID parameter record
PDRn.3	Float word	Tv	Derivative action time (in sec)	
PDRn.4	Float word	Kr	Feedback gain	
PDRn.5	Float word	T1	Time constant of fast feedback(in sec)	feedback par. record
PDRn.6	Float word	T2	Time constant of slow feedback (in sec)	
PDRn.7	Float word	HYS	Hysteresis of three-position switch	
PDRn.8	Float word	GUZ	Neutral zone	

DSUB Subtraction Double Word (SFB161)

Formal operand	Identi-fier	Meaning
DSUB		Operation (call)
DE1	MD addr.	<DE1> = M double word subtrahend
DE2	MD addr.	<DE2> = M double word minuend
DA	MD addr.	<DA> = M double word result
FM	Bit addr.	FM = 1: error

DT1 Diffenrentiator with Smoothing (SFB335)

Formal operand	Identi- fier	Number	Meaning
DT1			Operation (call)
STAT	Word addr.		Status control loop: -2 = Halt, -1 = Reset, 0 = Initial, 1 = Running
RST	Bit addr.		Control input for reset operating mode (”1” = Reset)
HALT	Bit addr.		Control input for halt operating mode (”1” = Halt)
PARA	PDT?		Data structure parameters (see below)
X	FWord addr.		Input
YRST	FWord addr.		Reset value output
YD	FWord addr.		Output differentiator without smoothing
Y	FWord addr.		Output differentiator with smoothing
AF	Bit addr.		AF = 1: error
WAF	Word addr.		Error code
VI	VIA?	16 byte	Data structure organization information

PDT

Element	Element type	Symbol suggestion	Meaning
PDTn.x			Data structure parameters of DT1, n = 50
PDTn.1	Float word	KD	Gain of differentiation
PDTn.2	Float word	T1	Delay time constant (in sec)

DWDN Convert BCD -40 Bit- to Double Word BCN (SFB150)

Formal operand	Identi- fier	Meaning
DWDN		Operation (call)
EF	Bit addr.	Enable
BK	Bit addr.	Smallest bit address of bit string
BL	Bit addr.	Last bit address of bit string
EV	Bit addr.	Sign input, EV=1: negative
WA	MD addr.	<WA> = Number converted into BCN code

DWND Convert Double Word BCN to BCD -40 Bit- (SFB148)

Formal operand	Identi- fier	Meaning
DWND		Operation (call)
EF	Bit addr.	Enable
WE	MD addr.	<WE> = Number converted into BCN code
BK	Bit addr.	Smallest bit address of bit string
BL	Bit addr.	Last bit address of bit string
AV	Bit addr.	Sign output, AV=1: negative

DWSAx Word Collector Double Words (SFB173-175)

Formal operand			Identi- fier	Meaning
DWSA4	DWSA8	DWS16		Operation (call)
EF	EF	EF	Bit addr.	Enable
DE1	DE1	DE1	MD addr.	1st input M double word
DE2	DE2	DE2	MD addr.	2nd input M double word
DE3	...	...	...	
DE4	DE8	DE16	MD addr.	Xth input M double word
DA	DA	DA	MD - addr.	Target M double word for DE1

DWVEx Word Distributor Double Word (SFB179-181)

Formal operand			Identi- fier	Meaning
DWVE4	DWVE8	DWV16		Operation (call)
EF	EF	EF	Bit addr.	Enable
DE	DE	DE	MD addr.	1st M double word of source word string
DA1	DA1	DA1	MD addr.	Target M double word for DE
DA2	DA2	DA2	MD addr.	Target M double word for DE+1
...	...	...	...	
DA4	DA8	DA16	MD addr.	Target M double word for DE+(x-1)

DZVR   Forwards-Backwards-Counter Double Word   (SFB104)

Formal operand	Identi-fier	Meaning
DZVR		Operation (call)
E	Bit addr.	Counter input, dynamic (0/1-edge)
ER	Bit addr.	Reset actual value: 0 → <WI>
EF	Bit addr.	Enable
EZ	Bit addr.	Count direction, EZ=0: forwards
WS	MD addr.	Setpoint value
AZ	Bit addr.	Previous signal state
WI	MD addr.	Actual value
AV	Bit addr.	Output sign of WI, AV=1: negative
WD	MD addr.	Differential value
AD	Bit addr.	Output sign of WD, AD=1: negative
AG	Bit addr.	Output “greater”, (<WI> > <WS>)
AA	Bit addr.	Output “equal”, (<WS> = <WI>)
AF	Bit addr.	AF = 1: error

EIN   Direct Input of a Pin String   (SFB187)

Formal operand	Identi-fier	Meaning
EIN		Operation (call)
EF	Bit addr.	Enable
EK	Bit addr.	Starting address of bit string (I/O)

EINR_POS Set up POS 102/112; A250, A500 as of BSW 6.0 (SFB62)

Formal-operand	Identi-fier	Number	Meaning
EINR_POS			Operation (call)
ER	Bit addr.		Reset
AKEN	Word addr.		Job code
SP / TN	TN addr.		for A250: <TN> = logical node number, 1 ... 31 for A500: <SP> = physical slot reference, 2 ... 160
ACHS	Word addr.		<ACHS> = Axis number ”0” = common axis operation ”1” = single axis operation, axis 1 ”2” = single axis operation, axis 2
SS	Bit addr.		Start/stop flag 0 → 1-edge: selection and start command for an operating mode, 1 → 0-edge: stop and deselect an operating mode
PARA	EPOA?		Data structure ”parameters of SFB EINR_POS”
ACCE	Bit addr.		Acknowledgement; ”1” = job accepted
AUFA	Bit addr.		”1” = code job error
RDY	Bit addr.		”1” = message job executed; during the data transmission between PLC and POS, <RDY> = 0
VI	VIPO?		Data structure organization information
AF	Bit addr.		Error code: AF = 1: error
WAF	Word addr.		Error code, <WAF> = error number

EPOA

Element	Element type	Symbol suggestion	Meaning
EPOAn			Data structure ”parameters of the SFB EINR_POS”, n = 1, ... , 64
EPOAn.1	Word	NRBTR	<> = Number of selected operating mode ”0” = no effect on POS, but error message ”1” = search for reference point (only valid for POS 102) ”2” = manual input
EPOAn.2	Bit	BE_KE	<> = unit definition for common and single axis operation, ”0” = absolute dimension ”1” = incremental dimension
EPOAn.3	Word	OVERRIDE	<> = override factor for manual input in common and single axis operation, can be set in % (0 ... 100)
EPOAn.4	Float word	VB_SOLL	<> = setpoint rate of contouring travel for manual input in common axis operation
EPOAn.5	Float word	EB1_SOLL	<> = setpoint speed of axis 1 for manual input in single axis operation
EPOAn.6	Float word	EB2_SOLL	<> = setpoint speed of axis 2 for manual input in single axis operation
EPOAn.7	Float word	P1_SOLL	<> = setpoint position of axis 1 for manual input in common and single axis operation
EPOAn.8	Float word	P2_SOLL	<> = setpoint position of axis 2 for manual input in common and single axis operation
EPOAn.9 ... EPOAn.15			Reserved

EWM Signal First-Up Value (SFB117)

Formal operand	Identi-fier	Meaning
EWM		Operation (call)
EF	Bit addr.	Enable
QF	Bit addr.	Acknowledgement of following events
QE	Bit addr.	Acknowledgement of first event
ZW	MW addr.	<ZW> = current states of the events
SE	MW addr.	<SE> = Value of first event
SF	MW addr.	<SF> = Value of following events
SQ	MW addr.	<SQ> = All acknowledged but non-reset events
SM	MW addr.	<SM> = Event signals to be displayed
AM	Bit addr.	Signal for first-up event

EWMV Signal First-Up Value, Linkable (SFB118)

Formal operand	Identi-fier	Meaning
EWMV		Operation (call)
EF	Bit addr.	Enable
EV	Bit addr.	EV: = 1 first-up event is not evaluated
QF	Bit addr.	Acknowledgement of following events (follow-up signal)
QE	Bit addr.	Acknowledgement of first event (first-up signal)
ZW	MW addr.	<ZW> = Current states of the events
SE	MW addr.	<SE> = Value of first event
SF	MW addr.	<SF> = Value of following events
SQ	MW addr.	<SQ> = All acknowledged but non-reset events
SM	MW addr.	<SM> = Event signals to be displayed
AV	Bit addr.	Signal for first-up event (linkable)
AM	Bit addr.	Signal for first-up event
AH	Bit addr.	Signal for horn control (1 scan)

EX Exponential Function (SFB282)

Formal operand	Identi-fier	Meaning
EX		Operation (call)
EF	Bit addr.	Enable
GE	MF addr.	Exponent of input quantity
GA	MF addr.	Output quantity
AF	Bit addr.	AF=1: error
WAF	MW addr.	Error code

FEA Edge Detection 0 $\leftrightarrow$ 1 (SFB114)

Formal operand	Identi-fier	Meaning
FEA		Operation (call)
E	Bit addr.	Input
AZ	Bit addr.	Previous state of E
A	Bit addr.	Output

FLA Edge Detection 1 $\rightarrow$ 0 (SFB113)

Formal operand	Identi-fier	Meaning
FLA		Operation (call)
E	Bit addr.	Input
AZ	Bit addr.	Previous state of E
A	Bit addr.	Output

FLE Edge Detection 0 $\rightarrow$ 1 (SFB112)

Formal operand	Identi-fier	Meaning
FLE		Operation (call)
E	Bit addr.	Input
AZ	Bit addr.	Previous state of E
A	Bit addr.	Output

FRB Filler Register Bit Processing (SFB232)

Formal operand	Identi-fier	Number	Meaning
FRB			Operation (call)
VI	MW addr.	2	Internal organization information
RL	MW addr.		<RL> = Register length, bit field length
RR	Bit addr.		Standardize, reset input/output window
TE	Bit addr.		Dynamic input clock
E	Bit addr.		Register input
TA	Bit addr.		Dynamic output clock
ER	Bit addr.		ER = 1: skip oldest information
RA	Bit addr.		Start of register, lowest bit field address
RQ	MW addr.		<RQ> = current filler level (0 ... <RL>)
A	Bit addr.		Register output
AF	Bit addr.		AF = 1: error

FRW Filler Register Word Processing (SFB233)

Formal operand	Identi-fier	Number	Meaning
FRW			Operation (call)
VI	MW addr.	2	Internal organization information
RA	MW addr.		Start of register, lowest word address
RL	MW addr.		<RL> = Register length, word field length: 1...
RB	MW addr.		<RB> = Register width, number bit, 1...8
RR	Bit addr.		Standardize, reset input/output window
TE	Bit addr.		Dynamic input clock
WE	MW addr.		Register input
TA	Bit addr.		Dynamic output clock
ER	Bit addr.		ER = 1: skip oldest information
RQ	MW addr.		<RQ> = current filler level, 0 ... <RL>
WA	MW addr.		Register output
AF	Bit addr.		AF = 1: error

GABS — Absolute Value Creation, Floating Point Word (SFB273)

Formal operand	Identi-fier	Meaning
GABS		Operation (call)
GE	MF addr.	Input quantity
AV	Bit addr.	Sign marker, AV=1: negative
GA	MF addr.	Output quantity
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GADD — Addition (SFB222)

Formal operand	Identi-fier	Meaning
GADD		Operation (call)
GE1	MF addr.	Summand 1
GE2	MF addr.	Summand 2
GA	MF addr.	Result
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GAEM Change Event, Floating Point Word (SFRB293)

Formal operand	Identi-fier	Meaning
GAEM		Operation (call)
EF	Bit addr.	Enable
GE	MF addr.	Input quantity
GUZ	MF addr.	Neutral zone
VI2	MF addr.	Internal organization information $\hat{=}$ GE_{old}
AA	Bit addr.	Event output
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GAEQ Equivalence -Compare- (SFB226)

Formal operand	Identi-fier	Meaning
GAEQ		Operation (call)
EF	Bit addr.	Enable
GE1	MF addr.	Input quantity 1
GE2	MF addr.	Input quantity 2
GUZ	MF addr.	Neutral zone (UZ)
AK	Bit addr.	<GE1> < <GE2> outside UZ
AA	Bit addr.	<GE1> = <GE2> inside UZ (+/-GUZ)
AG	Bit addr.	<GE1> > <GE2> outside UZ
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GAWA2 Floating Point Analog Value Output with MWA 012 (SFB257)

Formal operand	**Identi-fier**	**Meaning**
GAWA2		Operation (call)
EF	Bit addr.	Block enable
ER	Bit addr.	Reset
EP	Bit addr.	Output bipolar: EP = 0 Output unipolar: EP = 1
EW	Bit addr.	Set output area - without suppressed zero point (without open-circuit monitoring): EW = 0: 0 ... 10 V / -10 ... +10 V 0 ... 20 mA / -20 ... +20 mA - with suppressed zero point (with open-circuit monitoring) EW = 1: 2 ... 10 V 4 ... 20 mA
EBUA	Bit addr.	Behavior of outputs when monitoring time GTUE has expired. 0 = outputs go to 0 mA or 0 V (4 mA output for open-circuit monitoring is not possible) 1 = outputs retain old value
SP	MW addr.	<SP> = PEAB slot location of MWA 012 2 - 160 corresponding to entry in the equipment list
KA	MW addr.	<KA> = First channel to be output (channel 1 - 8)
KN	MW addr.	<KN> = Number of channels to be output (Number 1 - 8)
GTUE	MF addr.	<GTUE> = Monitoring time in sec from 0.2 ... 25.2 sec
GE	MF addr.	<GE> = 1st value to be output
GSKA	MF addr.	<GSKA> = Start of scale
GSKE	MF addr.	<GSKE> = End of scale
AF	Bit addr.	Error marker, AF = 1 if error
WAF	MW addr.	<WAF> = Error code

GAWA3 Analog Value Output to the MWA 16PN -Floating Point- (SFB218)

Formal operand	Identi- fier	Meaning
GAWA3		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Control input for reset
EW	Bit addr.	Set measuring range for BA unipolar
SP	MW addr.	<SP> = Slot reference of MWA 16PN, 2 ... 160
KA	MW addr.	<KA> = First channel to be read out, 1 ... 16
KN	MW addr.	<KN> = Number of channels to be read, 1 ... 17-<KA>
GE	MF addr.	<GE> = Digital value of 1st analog value to be output (channel)
GSKA	MF addr.	Start of scale (= offset)
GSKE	MF addr.	End of scale
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GAWA8 Analog Value Output with DAU 104/108 (Modnet 1/SFB) (SFB252)

Formal operand	Identi- fier	Meaning
GAWA8		Operation (call)
EF	Bit addr	Block enable
ER	Bit addr.	Reset
EW	Bit addr.	Reserved
SP	MW addr.	<SP> = Slot location; 1 or 2 ... 160, depending on ALU type used, corresponds to entry in the equipment list
KA	MW addr.	<KA> = First channel to be output (channel 1 ... 4 or 1 ... 8)
KN	MW addr.	<KN> = Number of channels to be output (Number 1 ... 4 or 1 ... 8)
GE	MF addr.	<GE> = 1st value to be output
GSKA	MF addr.	<GSKA> = Start of scale
GSKE	MF addr.	<GSKE> = End of scale
AF	Bit addr.	Error marker, AF = 1 if error
WAF	MW addr.	<WAF> = Error code

GAWE1 Analog Value Input from the ADU S9 -Floating Point- (SFB217)

Formal operand	Identi-fier	Meaning
GAWE1		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Control input for reset
SP	MW addr.	<SP> = Slot reference of ADU S9, 2 ... 160
KA	MW addr.	<KA> = First channel to be read in , 1 ... 256 *)
KN	MW addr.	<KN> = Number of channels to be read in, 1...255 *)
GSKA	MF addr.	Start of scale (= offset)
GSKE	MF addr.	End of scale
GOG	MF addr.	Upper limit of input quantity (in %)
GUG	MF addr.	Lower limit of input quantity (in %)
GA	MF addr.	1st output quantity (address of 1st channel)
AO	Bit addr.	Marker, upper limit GA violated
AU	Bit addr.	Marker, lower limit GA violated
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

*) corresponding to number of equipped channels

GAWE2 Floating Point Analog Value Input with ADU S12 (SFB256)

Formal operand	Identi-fier	Meaning
GAWE2		Operation (call)
EF	Bit addr	Block enable
ER	Bit addr.	Reset
EP	Bit addr.	Measuring range bipolar EP=0: -20 ... +20 mA / -10 ... +10 V Measuring range unipolar EP=1: 0 ... +20 mA / 0 ... +10 V
EW	Bit addr.	Set measuring range - without suppressed zero point (without open-circuit monitoring) EW=0: 0 ... 20 mA / 0 ... 10 V -20 ... +20 mA / -10 ... +10 V - with suppressed zero point (with open-circuit monitoring) EW=1: 4 ... 20 mA / 2 ... 10 V
SP	MW addr.	<SP> = PEAB slot 2 - 160
KA	MW addr.	<KA> = First channel to be read in (channel 1 - 14); channels 15 and 16 not allowed
KN	MW addr.	<KN> = Number of channels to be read (Number 1 - 14)
GSKA	MF addr.	<GSKA> = Start of scale
GSKE	MF addr.	<GSKE> = End of scale
GOG	MF addr.	<GOG> = Upper limit
GUG	MF addr.	<GUG> = Lower limit
GA	MF addr.	<GA> = 1st converted measured value (address of 1st channel)
AO	Bit addr.	AO = 1: Upper limit violated
AU	Bit addr.	AU = 1: Lower limit violated
WAG	MW addr.	<WAG> = Alarm if boundary violation
AF	Bit addr.	Error marker, AF = 1 if error
WAF	MW addr.	<WAF> = Error status

GAWE4 Floating Point Analog Value Input AEM 2511, EMU 2610 (SFB265)

Formal operand	Identi- fier	Number	Meaning
GAWE4			Operation (call)
EF	Bit addr.		EF = 1: enable
ER	Bit addr.		Control input for reset
SP	MW addr.		<SP> = Slot of AEM 2511 2...160
KA	MW addr.		<KA> = First channel to be read in 1...256 *)
KN	MW addr.		<KN> = Number of channels to be read 1...255 *)
GSKA	MF addr.		Start of scale (=offset)
GSKE	MF addr.		End of scale
GOG	MF addr.		Upper limit of input quantity (in %)
GUG	MF addr.		Lower limit of input quantity (in %)
EP	MW addr.		Operating mode switchover (BA), <EP>=0: bipolar, <EP>=1: unipolar
EW	MW addr.		Set measuring range for BA unipolar (EP=1), <EW>=0: without open-circuit monitoring, <EW>=1: with open-circuit monitoring
EV	MW addr.		Gain switchover <EV>=0: V=1, <EV>=1: V=8
GA	MF addr.		First output quantity (address of 1st channel)
AO	Bit addr.		AO = 1: upper limit GA exceeded
AU	Bit addr.		AU = 1: lower limit GA exceeded
AF	Bit addr.		AF=1: error
WAF	MW addr.		Error code

*) corresponding to number of equipped channel

GAWE6 Analog Value Input ADU 115/116, DAU 104, Modnet 1/SFB (SFB251)

Formal operand	**Identi-fier**	**Meaning**
GAWE6		Operation (call)
EF	Bit addr.	Block enable
ER	Bit addr.	Reset
EP	Bit addr.	- EP = 0: measuring range bipolar - EP = 1: measuring range unipolar
EW	Bit addr.	Set measuring range - EW = 0: without suppressed zero point (without open-circuit monitoring) - EW = 1: with suppressed zero point (with open-circuit monitoring)
EV	Bit addr.	Without function
SP	MW addr.	<SP> = Slot; 1 or 2 ... 160 depending on ALU used, corresponding to entry in the equipment list
KA	MW addr.	<KA> = First channel to be read in (channel 1 ... 16 or 8 for DAU 104)
KN	MW addr.	<KN> = Number of channels to be read (number 1 ... 16 or 8 for DAU 104)
GSKA	MF addr.	<GSKA> = Start of scale
GSKE	MF addr.	<GSKE> = End of scale
GOG	MF addr.	<GOG> = Upper limit
GUG	MF addr.	<GUG> = Lower limit
GA	MF addr.	<GA> = 1st converted measured value (address of 1st channel)
AO	Bit addr.	AO = 1: Upper limit violated
AU	Bit addr.	AU = 1: Lower limit violated
AF	Bit addr.	Error marker, AF = 1 if error
WAF	MW addr.	<WAF> = Error code

GAWS Analog Value Switch (SFB227)

Formal operand	Identi-fier	Meaning
GAWS		Operation (call)
EF	Bit addr.	Enable
EA	Bit addr.	Select input GE1 or GE2
GE1	MF addr.	Input quantity 1
GE2	MF addr.	Input quantity 2
GA	MF addr.	Output quantity
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GBGRZ Analog Value Limiter, Floating Point Word (SFB290)

Formal operand	Identi-fier	Meaning
GBGRZ		Operation (call)
EF	Bit addr.	Enable
GE	MF addr.	Input quantity
GOG	MF addr.	Upper limit
GUG	MF addr.	Lower limit
GA	MF addr.	Output quantity
AO	Bit addr.	Marker upper limit
AU	Bit addr.	Marker lower limit
SM	Bit addr.	Group signal, limit reached
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GDIV Division (SFB225)

Formal operand	Identi-fier	Meaning
GDIV		Operation (call)
GE1	MF addr.	Dividend
GE2	MF addr.	Divisor
GA	MF addr.	Result
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GDW1 Convert Floating Point Word to Double Word (SFB231)

Formal operand	Identi-fier	Meaning
GDW1		Operation (call)
EF	Bit addr.	Enable
GE	MF addr.	Floating point input quantity
DWA	MD addr.	Output double word
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GIW1 Convert Floating Point Word to Word (SFB230)

Formal operand	Identi-fier	Meaning
GIW1		Operation (call)
EF	Bit addr.	Enable
GE	MF addr.	Floating point input quantity
WA	MW addr.	Integer output word
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GKSAx Word Collector Floating Point Words (SFB211-213)

Formal operand			Identi-fier	Meaning
GKSA4	GKSA8	GKS16		Operation (call)
EF	EF	EF	Bit addr.	Enable
GE1	GE1	GE1	MF addr .	1st input floating point word
GE2	GE2	GE2	MF addr.	2nd input floating point word
GE3	....	....	....	
GE4	GE8	GE16	MF addr.	Xth input floating point word
GA	GA	GA	MF addr.	Target floating point word for GE1

GKVEx Word Distributor Floating Point Words (SFB214-216)

Formal operand			Identi-fier	Meaning
GKVE4	GKVE8	GKV16		Operation (call)
EF	EF	EF	Bit addr.	Enable
GE	GE	GE	MF addr.	1st floating point word of source word string
GA1	GA1	GA1	MF addr.	Target floating point word for GE
GA2	GA2	GA2	MF addr.	Target floating point word for GE+1
.....	....	....	...	
GA4	GA8	GA16	MF addr.	Target floating point word for GE+(x-1)

GLAx Load Floating Point, Effect as corresponding LAx (SFB207-210)

Formal operand	Identi-fier	Meaning	
GLA1		Operation (call)	
EF	Bit addr.	Enable	
R2	MW addr.	Source = constant	
R1	MF addr.	Target	see configuration
GLA2		Operation (call)	
EF	Bit addr.	Enable	
R2	MF addr.	Source	see configuration
R1	MF addr.	Target	see configuration
GLA4		Operation (call)	
EF	Bit addr.	Enable	
R2	MF addr.	Source	see configuration
R1	MW addr.	Target	see configuration
GLA5		Operation (call)	
EF	Bit addr.	Enable	
R2	MW addr.	Source	see configuration
R1	MF addr.	Target	see configuration

GMAXI   Maximum Value Selection, Floating Point Word   (SFB287)

Formal operand	Identi-fier	Meaning
GMAXI		Operation (call)
EF	Bit addr.	Enable
GE1	MF addr.	Input quantity 1
GE2	MF addr.	Input quantity 2
GE3	MF addr.	Input quantity 3
GE4	MF addr.	Input quantity 4
GA	MF addr.	Output quantity, selected maximum value
A1	Bit addr.	Selection marker 1
A2	Bit addr.	Selection marker 2
A3	Bit addr.	Selection marker 3
A4	Bit addr.	Selection marker 4
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GMINI   Minimum Value Selection, Floating Point Word   (SFB286)

Formal operand	Identi-fier	Meaning
GMINI		Operation (call)
EF	Bit addr.	Enable
GE1	MF addr.	Input quantity 1
GE2	MF addr.	Input quantity 2
GE3	MF addr.	Input quantity 3
GE4	MF addr.	Input quantity 4
GA	MF addr.	Output quantity, selected minimum value
A1	Bit addr.	Selection marker 1
A2	Bit addr.	Selection marker 2
A3	Bit addr.	Selection marker 3
A4	Bit addr.	Selection marker 4
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GMUL Multiplication (SFB224)

Formal operand	Identifier	Meaning
GMUL		Operation (call)
GE1	MF addr.	Multiplicand
GE2	MF addr.	Multiplicator
GA	MF addr.	Result
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GNEG Sign Reversal (Negation), Floating Point Word (SFB272)

Formal operand	Identifier	Meaning
GNEG		Operation (call)
EF	Bit addr.	Enable
EV	Bit addr.	Conversion enable
GE	MF addr.	Input quantity
GA	MF addr.	Output quantity (result)
AF	Bit addr.	AF=1: error
WAF	MW addr.	Error code

GPGON Interpolate Polygon, Floating Point Word (SFB289)

Formal operand	Identifier	Meaning
GPGON		Operation (call)
EF	Bit addr.	Enable
WEN	MW addr.	Starting address data field, number of interpolation points (see configuration)
GE	MF addr.	Input quantity
GA	MF addr.	Output quantity
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GQAD1 Square, Floating Point Word (SFB274)

Formal operand	Identi- fier	Meaning
GQAD1		Operation (call)
GE	MF addr.	Input quantity
AV	Bit addr.	Sign marker, AV=1: negative
GA	MF addr.	Output quantity
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GQAD2 Square with Sign, Floating Point Word (SFB275)

Formal operand	Identi- fier	Meaning
GQAD2		Operation (call)
GE	MF addr.	Input quantity
GA	MF addr.	Output quantity
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GRAD1 Root Extraction, Floating Point Word (SFB270)

Formal operand	Identi- fier	Meaning
GRAD1		Operation (call)
GE	MF addr.	Input quantity (radicand)
AV	Bit addr.	Sign of radicand (bit output), AV=1: negative
GA	MF addr.	Result
AF	Bit addr.	AF=1: error
WAF	MW addr.	Error code

GRAD2 Root Extraction with Sign, Floating Point Word (SFB271)

Formal operand	Identi-fier	Meaning
GRAD2		Operation (call)
GE	MF addr.	Input quantity (radicand)
GA	MF addr.	Result
AF	Bit addr.	AF=1: error
WAF	MW addr.	Error code

GRZMH Limit Signal with Hysteresis, Floating Point Word (SFB294)

Formal operand	Identi-fier	Meaning
GRZMH		Operation (call)
EF	Bit addr.	Enable
GE	MF addr.	Input quantity
GOG	MF addr.	Upper limit
GHO	MF addr.	Hysteresis upper limit
GUG	MF addr.	Lower limit
GHU	MF addr.	Hysteresis lower limit
AO	Bit addr.	Marker, upper limit reached
AU	Bit addr.	Marker, lower limit reached
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GSPM Peak Value Signal, Floating Point Word (SFB292)

Formal operand	Identi-fier	Meaning
GSPM		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Reset input for upper/lower starting value
GE	MF addr.	Floating point input quantity
GERO	MF addr.	Upper starting value
GERU	MF addr.	Lower starting value
GAO	MF addr.	Upper peak value (storing)
GAU	MF addr.	Lower peak value (storing)
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GSUB Subtraction (SFB223)

Formal operand	Identi-fier	Meaning
GSUB		Operation (call)
GE1	MF addr.	Summand 1
GE2	MF addr.	Summand 2
GA	MF addr.	Result
AF	Bit addr.	AF = 1 if error
WAF	MW addr.	Error code

GVERH Ratio Formation/Line Equation (SFB285)

Formal operand	Identi-fier	Number	Meaning
GVERH			Operation (call)
EF	Bit addr.		EF = 1: Enable
GE	MF addr.		<GE> = Input quantity (value X-axis)
GKA	MF addr.		<GKA> = Gain (line gradient)
GKB	MF addr.		<GKB> = Zero point offset Y-axis (constant)
GA	MF addr.		<GA> = Output quantity (value Y-axis)
AF	Bit addr.		AF = 1: error
WAF	MW addr.		Error code

GVORL Creation of Previous Load (reversal of neutral zone) (SFB288)

Formal operand	Identi-fier	Meaning
GVORL		Operation (call)
EF	Bit addr.	Enable
GE	MF addr.	Input quantity
GK	MF addr.	Gain GK = dGA / dGE
GKA	MF addr.	Offset for GE < 0
GKB	MF addr.	Offset for GE $\geq$ 0
GA	MF addr.	Output quantity
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

GWV Limit Comparison (SFB182)

Formal operand	Identi- fier	Meaning
GWV		Operation (call)
EF	Bit addr.	Enable
EQ	Bit addr.	Acknowledgement
MWA	MW addr.	Address of 1st measured value
MWE	MW addr.	Address of last measured value
OG	MW addr.	<OG> = Upper limit
UG	MW addr.	<UG> = Lower limit
GWO	MW addr.	<GWO> = Code for upper limit violation
GWU	MW addr.	<GWU> = Code for lower limit violation
AO	Bit addr.	AO = 1: Upper limit violation
AU	Bit addr.	AU = 1: Lower limit violation

HAD Manual Operation (SFB250)

Formal operand	Identi- fier	Meaning
HAD		Operation (call)
EF	Bit addr.	Enable
SP	MW addr.	Slot reference POS
EP	Bit addr.	(Positive) direction
EM	Bit addr.	(Negative) direction
FR	Bit addr.	Reset error
DWV	MD addr.	Feed
WAF	MW addr.	Error code

HAND_POS Manual Control POS 102/112; A500 as of BSW 6.0 (SFB63)

Formal operand	Identi- fier	Number	Meaning
HAND_POS			Operation (call)
ER	Bit addr.		”1” = Reset
AKEN	Word addr.		Job code
TN / SP	TN addr.		for A250: <TN> = logical node number, 1 ... 31 for A500: <SP> = physical slot reference, 2 ... 160
ACHS	Word addr.		<ACHS> = Axis number ”0” = common axis operation ”1” = single axis operation, axis 1 ”2” = single axis operation, axis 2
SS	Bit addr.		Start/stop flag
PARA	HPOA?		Data structure ”parameters for SFB HAND_POS”
ACCE	Bit addr.		Acknowledgement, ”1” = job accepted
AUFA	Bit addr.		”1” = job error
VI	VIPO?		Data structure organization information
AF	Bit addr.		Error marker, AF = 1: error
WAF	Word addr.		Error code, <WAF> = Error number

HPOA

Element	Element type	Symbol suggestion	Meaning
HPOAn			Data structure ”parameters for SFB HAND_POS” n = 1, ..., 64
HPOAn.1	Word	NRBTR_H	<> = Number of selected operating mode ”0” = no effect on the POS except for error message ”1” = manual control
HPOAn.2	Word	OVERR_H	Override factor for manual control in % (0 ... 100)
HPOAn.3	Bit	F_POSX	”1” = Drive axis 1 in positive direction
HPOAn.4	Bit	F_NEGX	”1” = Drive axis 1 in negative direction
HPOAn.5	Bit	F_POSY	”1” = Drive axis 2 in positive direction
HPOAn.6	Bit	F_NEGY	”1” = Drive axis 2 in negative direction
HPOAn.7	Float word	VX_H	<> = Setpoint speed axis 1; Floating point word is always positive. The direction is defined by the preceding bits. The movement continues as long as SS = ”1”.
HPOAn.8	Float word	VY_H	<> = Setpoint speed axis 2; see HPOAn.7
HPOAn.9 ... HPOAn.16			Reserved

IB Integrator with Limit (SFB345)

Formal operand	Identi-fier	Number	Meaning
IB			Operation (call)
STAT	Word addr.		Status control loop: -2 = Halt, -1 = Reset, 0 = Initial, 1 = Running
RST	Bit addr.		Control input for operating mode Reset ("1" = Reset)
HALT	Bit addr.		Control input for operating mode Halt ("1" = Halt)
PARA	PIB?		Data structure parameters (see below)
X	FWord addr.		Input
YRST	FWord addr.		Reset value output
Y	FWord addr.		Output
YAO	Bit addr.		Output reached upper limit (YAO = "1")
YAU	Bit addr.		Output reached lower limit (YAU = "1")
AF	Bit addr.		AF = 1: error
WAF	Word addr.		Error code
VI	VIA?	16 byte	Data structure organization information

PIB

Element	Element type	Symbol suggestion	Meaning
PIBn.x			Data structure parameters of IB
PIBn.1	Float word	KI	Integration gain
PIBn.2	Float word	OG	Upper limit
PIBn.3	Float word	UG	Lower limit

ICR Incrementer (SFB154)

Formal operand	Identi-fier	Meaning
ICR		Operation (call)
EF	Bit addr.	Enable
WE	MW addr.	Input M word
WA	MW addr.	Output M word

IDS (SFB490)
IDST (SFB491)

☞ **Note** The documention about SFB490 and SFB491 can be obtained from AEG Weinheim, Information from Mr. Krick, Tel. 06201/603320.

IGW1 Convert Word to Floating Point Word (SFB228)

Formal operand	Identi-fier	Meaning
IGW1		Operation (call)
EF	Bit addr.	Enable
WE	MW addr.	Integer input word
GA	MF addr.	Floating point output quantity

IMA Transmit Maschine Parameters POS → PLC (SFB246)

Formal operand	Identi-fier	Meaning	
IMA		Operation (call)	
EF	Bit addr.	Enable	
SP	MW addr.	Slot refernce POS	
WB	MW addr.	Memory area	PLC
WA	MW addr.	Offset	
VI2	MW addr.	Internal organization information	
RDY	Bit addr.	Ready signal	

INV Invert Word (SFB101)

Formal operand	Identi-fier	Meaning
INV		Operation (call)
EF	Bit addr.	Enable
DW	MW addr.	<DW>=Input value
DK	MW addr.	<DK>=<DW> inverted

IPR Program Security POS → PLC (SFB243)

Formal operand	Identi-fier	Meaning	
IPR		Operation (call)	
EF	Bit addr.	Enable	
SP	MW addr.	Slot reference POS	
WB	MW addr.	Memory area	PLC
WA	MW addr.	Offset	
VI2	MW addr.	Internal organization information	
RDY	Bit addr.	Ready signal	

IST Receive Actual Position (SFB247)

Formal operand	Identi-fier	Meaning
IST		Operation (call)
EF	Bit addr.	Enable
SP	MW addr.	Slot reference POS
RA	Bit addr.	Approach reference point
PO	Bit addr.	In position
WBA	MW addr.	<WBA> = operating mode; 1: approach reference point; 2: position presetting; 4: single record; 16: manual control; 32: manual input; 64: automatic; 128: memory input/output
WST	MW addr.	Status signal input/output
DWI	MD addr.	Actual position
AF	Bit addr.	AF = 1: error
WAF	MW addr.	Error code

ISTD_POS Actual Data of POS 102/112; A500 as of BSW 6.0 (SFB61)

Formal operand	Identi-fier	Number	Meaning
ISTD_POS			Operation (call)
ER	Bit addr.		”1” = reset
AKEN	Word addr.		<AKEN> = job code
TN / SP	TN addr.		for A250: <TN> = logical node number, 1 ... 31; for A500: <SP> = physical slot reference, 2 ... 160
ACHS	Word addr.		<ACHS> = axis number. The effect of AUQU, GRUN, HLT, MOD, QUIT, STP refers to this ”0” = common axis operation ”1” = single axis operation, axis 1 ”2” = single axis operation, axis 2
HBUM	Word addr.		<HBUM> = Switchover between the main operating modes, ”1” = single axis operation, ”0” = common axis operation
MOD	Bit addr.		Switchover between motion mode and simulation mode ”0” = motion mode, ”1” = simulation mode
SS	Bit addr.		Start/stop flag
STP	Bit addr.		”1” = cause reaction ”Stop” on the POS
HLT	Bit addr.		”1” = cause reaction ”Halt” on the POS
GRUN	Bit addr.		”1” = cause basic setting, ”0” = operating mode can be selected
AUQU	Bit addr.		”1” = errors occurring on POS are automatically acknowledged
QUIT	Bit addr.		errors occurring on POS are acknowledged with 0–>1-edge
ACCE	Bit addr.		”1” = acknowledgement job accepted
AUFA	Bit addr.		”1” = code for job error
RDY	Bit addr.		”1” = signal job executed; during the data transmission between PLC and POS, <RDY> = 0
ISVB	VBPA?		Data structure ”module-dependent actual data for POS”
ISB1	AXPA?		Data structure ”axis-dependent actual data for axis 1”
ISB2	AXPA?		Data structure ”axis-dependent actual data for axis 2”
VI	VIPO?		Data structure organization information
AF	Bit addr.		Error marker, AF = 1: error
WAF	Word addr.		Error code, <WAF> = Error number

VBPA

Element	Element type	Symbol suggestion	Meaning
VBPAn			Data structure ”module-dependent actual data for POS”, n = 1, ..., 64
VBPAn.1	Word	POSTYP	<> = type of POS inserted (102 ≙ POS102, 112 ≙ POS 112)
VBPAn.2	Word	HAUPTBTR	<> = active main operating mode ”0” = common axis operation, ”1” = single axis op.
VBPAn.3	Word	BEWMODE	<> = current operating mode in common axis op. -2: POS cannot carry out any motion, since its configuration data is missing -1: module must make search for reference point (only POS 102) 0: basic setting 1: search for reference point 3: manual input 4: manual control or teach-in (teach-in currently only possible with PADT) 6: program automatic 7: program automatic single step 8: program automatic external (in preparation)
VBPAn.4	Word	ST_CASE	<> = last start mode of module: ”1” = cold start, ”2” = cold restart, ”3” = warm restart
VBPAn.5	Word	OR_BAHN	<> = current override factor in common axis operation, display in % (0 ... 100)
VBPAn.6	Float word	V_BAHN	<> = current rate of contouring travel (common axis operation); valid for common and single axis operation. The units correspond to the values set with the PADT in the POS presetting
VBPAn.7	Bit	MD_KOMPL	”1” = machine data on the POS complete
VBPAn.8	Bit	V24_ZUST	Write access rights (”1” = PLC, ”0” = PADT)
VBPAn.9	Bit	STOP_W	”1” = Stop signal has effect (common axis operation)
VBPAn.10	Bit	HALT_W	”1” = Halt signal has effect (common axis operation)
VBPAn.11	Bit	GRUND_W	”1” = Basic setting has effect (common axis operation)
VBPAn.12	Word	VP_NR	<> = Number of the current traversing program (common axis operation), 1 ... 65 ”0” means: no traversing program selected
VBPAn.13	Word	SATZ_NR	<> = Current record number of active traversing program (common axis operation)

Continued on next page

VBPA

Element	Element type	Symbol suggestion	Meaning
Continued			
VBPAn.14	Word	VP_STS	<> = Current traversing program status (common axis operation) 0: no traversing program selected 1: traversing program processed 2: traversing program selected, wait for start command 3: traversing program terminated with M30 4: traversing program terminated with M02 5: traversing program started with record select (dry run)
VBPAn.15	Bit	VP_HALT	”1” = Halt for current traversing program in effect (common axis operation)
VBPAn.16	Bit	SATZ_UNT	”1” = Record suppression in effect (common axis operation)
VBPAn.17	Bit	EINZ_STP	”1” = Single step operation in effect (common axis operation)
VBPAn.18	Bit	SIM_WIRK	”1” = Simulation mode in effect (common axis operation)
VBPAn.19	Word	TOOL_NR	<> = Number of effective tool correction memory for active tool correction (common axis operation), 1 ... 20 ”0” means: no tool correction memory selected
VBPAn.20	Word	AE1	Input value of analog input 1 (10 VDC = 32 000)
VBPAn.21	Word	AE2	Input value of analog input 2 (10 VDC = 32 000)
VBPAn.22	Bit	E1	Value of input 1 on the module
VBPAn.23	Bit	E2	Value of input 2 on the module
VBPAn.24	Bit	E3	Value of input 3 on the module
VBPAn.25	Bit	E4	Value of input 4 on the module
VBPAn.26	Bit	E5	Value of input 5 on the module
VBPAn.27	Bit	E6	Value of input 6 on the module
VBPAn.28	Bit	E7	Value of input 7 on the module
VBPAn.29	Bit	E8	Value of input 8 on the module
VBPAn.30	Bit	A1	Value of output 1 on the module
VBPAn.31	Bit	A2	Value of output 2 on the module
VBPAn.32	Bit	A3	Value of output 3 on the module
VBPAn.33	Bit	A4	Value of output 4 on the module
VBPAn.34	Bit	A5	Value of output 5 on the module
VBPAn.35	Bit	A6	Value of output 6 on the module

Continued on next page

VBPA

Element	Element type	Symbol suggestion	Meaning
Continued			
VBPAn.36	Bit	A7	Value of output 7 on the module
VBPAn.37	Bit	A8	Value of output 8 on the module
VBPAn.38	Word	LEB_VB	Scan counter, is incremented each time the data structure is executed. It shows whether POS is actively linked to the PLC and ready for service or in operation. All the actual data in the block are updated in the same scan.
VBPAn.39 ... VBPAn.49			Reserved

APXA

Element	Element type	Symbol suggestion	Meaning
AXPAn axis”,			Data structure ”axis-dependent actual data for n = 1 ... 128
AXPAn.1	Word	BEW_EB	<> = Current operating mode of the axis (single axis operation) -2: The axis cannot carry out any motion since its configuration data is missing -1: Axis must execute search for reference point (only for POS 102) 0: Basic setting 1: Search for reference point 2: Automatic control setting (in preparation 3: Manual input 4: Manual control or teach-in (teach-in currently only possible with PADT) 5: External follow-up operation 6: Program automatic 7: Program automatic single step 8: Program automatic external (in preparation)
AXPAn.2	Word	OVR_EB	<> = Current override factor for axis in single axis operation, display in % (0 ... 100)
AXPAn.3	Float word	V_ACHSE	<> = Current axis speed for common and single axis operation
AXPAn.4	Float word	S_ACHSE	<> = Current following error of axis
AXPAn.5	Float word	P_ACHSE	<> = Current position of axis

Continued on next page

AXPA

Element	Element type	Symbol suggestion	Meaning
Continued			
AXPAn.6	Bit	STOP_EB	”1” = Stop signal for the axis in effect (single axis operation)
AXPAn.7	Bit	HALT_EB	”1” = Halt signal for the axis in effect (single axis operation)
AXPAn.8	Bit	GRUND_EB	”1” = Basic setting for the axis in effect (single axis operation)
AXPAn.9	Word	VP_NR_EB	<> = Current no. of traversing program selected for the axis (single axis operation), 1 ... 65, ”0” means: no traversing program selected
AXPAn.10	Word	ST_NR_EB	<> = Current record number of traversing program selected for the axis (single axis operation)
AXPAn.11	Word	VP_ST_EB	<> = Current traversing program status (single axis operation) 0: no traversing program selected 1: traversing program processed 2: traversing program selected, wait for start command 3: traversing program terminated with M30 4: traversing program terminated with M02 5: traversing program started with record select (dry run)
AXPAn.12	Bit	VP_HT_EB	”1” = Halt for current traversing program in effect (single axis operation)
AXPAn.13	Bit	ST_UN_EB	”1” = Record suppression in effect (single axis operation)
AXPAn.14	Bit	VP_EZ_EB	”1” = Single step operation in effect (single axis operation)
AXPAn.15	Bit	SIM_EB	”1” = Simulation mode in efffect (single axis operation)
AXPAn.16	Word	TO_NR_EB	<> = Number of selected tool correction memory for active tool correction (single axis operation), 1 ... 20 ”0” means: no tool correction memory selected
AXPAn.17	Bit	REF_ACHS	”1” = Axis is referenced
AXPAn.18	Bit	REG_BERT	”1” = Controller ready signal for axis in effect
AXPAn.19	Bit	BREMS_AS	”1” = Brake for axis is released
AXPAn.20	Bit	REG_FREI	”1” = Control enable for axis in effect
AXPAn.21	Bit	FAHR_AS	”1” = Drive command for axis in effect
AXPAn.22	Bit	AX_IN_PO	”1” = Axis is in position
AXPAn.23	Bit	MAX_SCHL	”1” = Maximum following error of axis reached

Continued on next page

AXPA

Element	Element type	Symbol suggestion	Meaning
Continued			
AXPAn.24	Bit	R_SW_SW	”1” = Positive software end switch reached
AXPAn.25	Bit	L_SW_SW	”1” = Negative software end switch reached
AXPAn.26	Bit	R_HW_SW	”1” = Positive hardware end switch reached
AXPAn.27	Bit	L_HW_SW	”1” = Negative hardware end switch reached
AXPAn.28	Bit	FAHR_POS	”1” = Drive in positive direction not allowed
AXPAn.29	Bit	FAHR_NEG	”1” = Drive in negative direction not allowed
AXPAn.30	Bit	REG_SPER	”1” = Control inhibit for axis in effect
AXPAn.31	Bit	FLIEG_AU	”1” = Floating M function in effect
AXPAn.32	Word	FEHL_NR	<> = Current axis-dependent error number on POS (error number of last error not acknowledged is displayed)
AXPAn.33	Word	LEB_EB	Scan counter; is incremented each time the data structure is executed. It shows whether POS is actively linked to the PLC and ready for operation or in operation. All the actual data in the block are updated with the same scan.
AXPAn.34 ... AXPAn.63			Reserved

KOST KOS Control Block (SFB40)

Formal operand	Identi-fier	Number	Meaning
KOST			Operation (call))
EF	Bit addr.		Enable
KONR	Word addr.		<KONR> = KP number: KOS No. 1 ... 7
SPBI	Bit addr.	9	Start address inhibit bit
STBI	Bit addr.	9	Start address control bit
ZKBI	Bit addr.	126	Start address ZKA errort
KOWO	Word addr.	27	Start address KOS control
KOZK	Word addr.	4	Start address ZKA wrap around counter

KPID Complex PID-Controller (SFB300)

Formal operand	Identi-fier	Meaning
KPID		Operation (call)
STAT	Word addr.	Status control loop: -2 = Halt, -1 = Reset, 0 = Initial, 1 = Running
BT	BKID?	Data structure operating modes (see below)
PARA	PKID?	Data structure parameters (see below)
WE	FWord addr.	Setpoint value input for cascade
W	FWord addr.	Setpoint value input
X	FWord addr.	Actual value input
YHND	FWord addr.	Manual value manipulated variable
YRST	FWord addr.	Reset value manipulated variable
Z	FWord addr.	Disturbance input
AP	FWord addr.	Working point
XD	FWord addr.	Output error signal
Y	FWord addr.	Manipulated variable
MELD	MKID?	Data structure signals (see below)
AF	Bit addr.	AF = 1: error
WAF	Word addr.	Error code
VI	VIB?	Data structure organization information

BKID

Element	Element type	Symbol suggestion	Meaning
BKIDn.x			Data structure operating modes of KPID, n = 30
BKIDn.1	Bit	Reset	Control input for reset operating mode ("1" = Reset, Y = YRST)
BKIDn.2	Bit	Hand	Control input for manual operating mode ("1" = Hand, Y =YHND)
BKIDn.3	Bit	Halt	Control input for halt operating mode ("1" = Halt, Y unchanged)
BKIDn.4	Bit	Kaskade	Control input for cascade operating mode ("1" = cascade)
BKIDn.5	Bit	P-Ein	"1" = P-part on
BKIDn.6	Bit	I-Ein	"1" = I-part on
BKIDn.7	Bit	D-Ein	"1" = D-part on
BKIDn.8	Bit	EDUM	"1" = D-part to controlled variable, "0" = D-part to error signal
BKIDn.9	Bit	EBUM	"1" = Antiwindup Halt, "0" = Antiwindup Reset
BKIDn.10	Bit	STOS	"1" = Switchover, "0" = Bumpless changeover
BKIDn.11	Bit	YH-Nach	"1" = Influence YH

PKID

Element	Element type	Symbol suggestion	Meaning
PKIDn.x			Data structure parameters of KPID, n = 30
PKIDn.1	Float word	Kp	Proportional rate (gain)
PKIDn.2	Float word	Tn	Reset time (in sec)
PKIDn.3	Float word	Tv	Derivative action time (in sec)
PKIDn.4	Float word	T1	Delay time constants for D-part (in sec)
PKIDn.5	Float word	UZ	Neutral zone
PKIDn.6	Float word	KUZ	Gain reduction within UZ
PKIDn.7	Float word	VWH	Rate of change limit for W
PKIDn.8	Flaot word	VYH	Rate of change limit for YH
PKIDn.9	Float word	YOH	Upper limit
PKIDn.10	Float word	YUH	Lower limit
PKIDn.11	Float word	DYAW	Extend limit for antiwindup

MKID

Element	Element type	Symbol suggestion	Meaning
MKIDn.x			Data structure signals from KPIDR, n = 30
MKIDn.1	Bit	Reset	”1” = signal: KPID in reset operating mode
MKIDn.2	Bit	Hand	”1” = signal: KPID in manual operating mode
MKIDn.3	Bit	Halt	”1” = signal: KPID in halt operating mode
MKIDn.4	Bit	Auto	”1” = signal: KPID in automatic operating mode
MKIDn.5	Bit	Kaskade	”1” = signal: KPID in cascade operating mode
MKIDn.6	Bit	YAO	”1” = signal: Y reached upper limit
MKIDn.7	Bit	YAU	”1” = signal: Y reached lower limit

KPT — Complement Word -Reversal of Sign- (SFB151)

Formal operand	Identi-fier	Meaning
KPT		Operation (call)
EF	Bit addr.	Enable
DW	MW addr.	<DW> = Number whose sign is to be reversed
DK	MW addr.	<DK> = Result = <-DW>

LAx — Load from Source to Target (SFB135-140)

Formal operand	Identi-fier	Meaning
LA1, LA3		Operation (call)
EF	Bit addr.	Enable
R2	MW addr.	Source = constant
R1	MW addr.	Target (LA3: indirect target) see configuration
LA2, LA4...LA6		Operation (call)
EF	Bit addr.	Enable
R2	MW addr.	Source (LA5 and LA6: indirect source) see config.
R1	MW addr.	Target (LA4 and LA6: indirect target) see config.

LB500 Control Operation B500 → A500 (SFB261)

Formal operand	**Identi-fier**	**Meaning**
LB500		Operation (call)
EF	Bit addr.	Enable
LB	Bit addr.	Control bit
SB	Bit addr.	Transmit bit SEAB telegram
OW	MW addr.	<OW> = Offset target address word
OGW	MW addr.	<OGW> = Offset target address word
OB	MW addr.	<OB> = Offset target address bit
PV	MW addr.	<PV> = PV-No. of control process variables
UA	MW addr.	<UA> = Original address
ZA	MW addr.	<ZA> = Target address
AW	MW addr.	<AW> =Value of control process variables word
AGW	MF addr.	<AGW> = Value of control process variables floating point word
AB	Bit addr.	<AB> = Value of control process variables bit
KW	Bit addr.	Code word
KGW	Bit addr.	Code floating point word
KB	Bit addr.	Code bit
WAF	MW addr.	Error code
AF	Bit addr.	AF = 1: error

LBF Load Bit Field -Marker- (SFB127)

Formal operand	**Identi-fier**	**Meaning**
LBF		Operation (call)
EF	Bit addr.	Enable
WBU	MW addr.	<WBU>= Memory area of source data
WU	MW addr.	<WU> = Starting address of source data
WN	MW addr.	<WN> = Number of source data
WBZ	MW addr.	<WBZ>= Memory area of target data
WZ	MW addr.	<WZ> = Starting address of target data

LBS — Load Bit String -Indirect- (SFB128)

Formal operand	Identi-fier	Meaning
LBS		Operation (call)
EF	Bit addr.	Enable
WBKU	MW addr.	<WBKU> = Start of source marker string
WBLU	MW addr.	<WBLU> = End of source marker string
WBKZ	MW addr.	<WBKZ> = Start of target marker string
AF	Bit addr.	AF = 1: error

LBW — Load Bit String to Word (SFB131)

Formal operand	Identi-fier	Meaning
LBW		Operation (call)
BK	Bit addr.	Smallest bit address of bit string (corresponds to most significant bit)
BL	Bit addr.	Last bit address of bit string (corresponds to least significant bit)
WA	MW addr.	Output - M word

LDF — Load Data Field -Words- (SFB126)

Formal operand	Identi-fier	Meaning
LDF		Operation (call)
EF	Bit addr.	Enable
WBU	MW addr.	<WBU> = Memory area of source data
WU	MW addr.	<WU> = Starting address of source data
WN	MW addr.	<WN> = Number of source data
WBZ	MW addr.	<WBZ> = Memory area of target data
WZ	MW addr.	<WZ> = Starting address of target data

LDSG Load Segment (SFB263)

Formal operand	Identi-fier	Meaning
LDSG		Operation (call)
EF	Bit addr.	Enable
WSGU	MW addr.	<WSGU> = Segment no. of source data
WU	MW addr.	<WU> = Starting addr. of source data
WN	MW addr.	<WN> = Number of source data
WSGZ	MW addr.	<WSGZ> = Segment no. of target data
WZ	MW addr.	<WZ> = Starting address of target data

LEB Delete Bit Area (SFB119)

Formal operand	Identi-fier	Meaning
LEB		Operation (call)
EF	Bit addr.	Enable
BK	Bit addr.	Smallest bit address of bit string
BL	Bit addr.	Last bit address of bit string

LED Delete Double Word Area (SFB121)

Formal operand	Identi-fier	Meaning
LED		Operation (call)
EF	Bit addr.	Enable
DK	MD addr.	First M double word
DL	MD addr.	Last M double word

LEG Delete Floating Point Word Area (SFB206)

Formal operand	Identi-fier	Meaning
LEG		Operation (call)
EF	Bit addr.	Enable
GK	MF addr.	First floating point word
GL	MF addr.	Last floating point word

LEW Delete Word Area (SFB120)

Formal operand	Identi-fier	Meaning
LEW		Operation (call)
EF	Bit addr.	Enable
WK	MW addr.	First word
WL	MW addr.	Last word

LG Brigg's Logarithm (decadic) (SFB284)

Formal operand	Identi-fier	Meaning
LG		Operation (call)
EF	Bit addr.	Enable
GE	MF addr.	Input quantity
GA	MF addr.	Output quantity
AV	Bit addr.	Output sign, AV=1: negative
AF	Bit addr.	AF=1: error
WAF	MW addr.	Error code

LIN Linearize Measured Value (SFB186)

Formal operand	Identi-fier	Meaning
LIN		Operation (call)
EF	Bit addr.	Enable
TN	Bit addr.	TN = 1: Compute tangent
MA	MW addr.	Starting address of value table
ME	MW addr.	End address of value table
WRA	MW addr.	Starting address of measured values
WRE	MW addr.	End address of measured values
WP	MW addr.	Starting address of physical values of WRA ... WRE

LN Natural Logarithm (SFB283)

Formal operand	Identi- fier	Meaning
LN		Operation (call)
EF	Bit addr.	Enable
GE	MF addr.	Input quantity
GA	MF addr.	Output quantity
AV	Bit addr.	Input sign, AV=1: negative
AF	Bit addr.	AF=1: error
WAF	MW addr.	Error code

LWB Load Word to Bit String (SFB129)

Formal operand	Identi- fier	Meaning
LWB		Operation (call)
WE	MW addr.	Input - M word
BK	Bit addr.	Smallest bit address of bit string (corresponds to most significant bit)
BL	Bit addr.	Last bit address of bit string (corresponds to least significant bit)

MAP Transmit Machine Parameters PLC → POS (SFB245)

Formal operand	Identi- fier	Meaning
MAP		Operation (call)
EF	Bit addr.	Enable
SP	MW addr.	Slot reference POS
WB	MW addr.	Memory area PLC
WA	MW addr.	Offset
VI2	MW addr.	Internal organization information
RDY	Bit addr.	Ready signal

MARK (SFB221)

Formal operand	Identi- fier	Number	Meaning
MARK			Operation (call)
XOG	MF addr.		Upper limit
XOW	MF addr.		Upper warn value
XUW	MF addr.		Lower warn value
XUG	MF addr.		Lower limit
PI7M	MW addr.	7 Words	organization info., (P-parameter in signal file)
PI7B	MW addr.		PI7B of corresponding bar block

MUE Multiplication Word (SFB158)

Formal operand	Identi- fier	Meaning
MUE		Operation (call)
WE	MW addr.	<WE> = Factor
WE	MW addr.	<WE> = Factor
WA	MW addr.	<WA> = Result
AF	Bit addr.	AF = 1: error

MWB Mean Value Generation (SFB185)

Formal operand	Identi- fier	Meaning
MWB		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Reset
MW	MW addr.	<MW> = Current measured value
SU	MD addr.	<SU> = Sum of measured values
IZ	MW addr.	<IZ> = Number of added measured values
MI	MW addr.	<MI> = Mean value

NOK Transmission NOK 116 → A500 (SFB10)

Formal operand	Identi-fier	Number	Meaning
NOK			Operation (call)
EF	Bit addr.		Enable
ER	Bit addr.		Reset block
SP	MW addr.		Slot (must agree with entry in the equipment list)
VI2	MW addr.	2	Internal organization information
DREH	MW addr.		Current speed (0 ... 800)
HWWI	MW addr.		Current main shaft angle (0 ... 359°)
HUBI	MW addr.		Current stroke (0 ... 999 mm)
NACH	MW addr.		Overtravel angle (0 ... 359°)
IST1	MD addr.		Actual number of pieces 1 (0 ... 999 999)
IST2	MD addr.		Actual number of pieces 2 (0 ... 999 999)
AF	Bit addr.		AF = 1: error
WAF	MW addr.		<WAF> = Error code

NOKI Intialization NOK 116 (SFB11)

Formal operand	Identi-fier	Number	Meaning
NOKI			Operation (call)
EF	Bit addr.		Enable
ER	Bit addr.		Reset block
UEBR	Bit addr.		Define transmission direction: 0 = NOK 116 → PLC 1 = PLC → NOK 116
ES	Bit addr.		Access inhibit for V.24-interface of NOK 116: 0 → access inhibited 1 → access possible
EQ	Bit addr.		EQ: 0 → 1: Short circuit acknowledgement (only reacts to edge). The bit is reset again after the transmission to the NOK 116.
EBUA	MW addr.		Behavior of NOK 116 after expiration of monitoring time: 0 = retain function of NOK 116 1 = NOK 116 stops in upper top dead center
TUE	MW addr.		Monitoring time: <TUE> = 1 ... 255; step width 25 msec, presetting of NOK 116 = 100 ≙ 2.5 sec TUE_{max} = 255 x 25 msec = 6.375 sec
SP	MW addr.		Slot (must agree with the entry in the equipment list)

Continued on next page

Formal operand	Identi- fier	Number	Meaning
NOKI Continued			
EFB1	Bit addr.		Block enable 1st block (tool data)
IST1	MD addr.		Actual number of pieces 1 (0 ... 999 999)
IST2	MD addr.		Actual number of pieces 2 (0 ... 999 999)
END1	MD addr.		End number of pieces 1 (0 ... 999 999). When the end number of pieces is reached, the correspon-ding output becomes 1. The press is not halted.
END2	MD addr.		End number of pieces 2 (0 ... 999 999). When the end number of pieces is reached, the correspon-ding output becomes 1. The press is not halted.
SCH1	MW addr.		Step width multiplicator 1; 1 ... 99 $\triangleq$ Production of n pieces per clamping stroke 0 $\triangleq$ counter inactive
SCH2	MW addr.		Step width multiplicator 2; 1 ... 99 $\triangleq$ Production of n pieces per clamping stroke 0 $\triangleq$ counter inactive
HUBW	MW addr.		Required stroke (0 ... 999 mm)
HUBO	MW addr.		Actual stroke offset (0 ... 359^o)
NK	MW addr.		Cam number as of cam 1 (1 ... 16)
STAT	MW addr.		Status field. One word per cam (number of cams is defined by NK, i.e. max. 16 words). The cam status is defined according to the following table: 0 = cam inactive 1 = angle/angle cams 2 = angle/time cams 3 = as for 1 4 = as 0 5 = angle/angle cams with lead correction 6 = angle/time cams with lead correction No other values are permitted.
NOKA	MW addr.		Cam starting field. One word per cam (number of cams is defined by NK, i.e. max 16 words). The contents of the particular words define the cam starting angle (0 ... 359^o).
NOKE	MW addr.		Cam end field/cam time. One word per cam (number of cams is defined by NK, i.e. max 16 words). The contents of the particular words define the cam end angle or cam time (0 ... 359^o, 0 ... 999 msec).
NOKV	MW addr.		Cam lead time. One word per cam (number of cams is defined by NK, i.e. max 16 words). The contents of the particular words define the cam lead time (0 ... 250 msec).

Continued on next page

Formal operand	Identi- fier	Number	Meaning
NOKI Continued			
EFB2	Bit addr.		Block enable 2nd block (machine data)
WB	MW addr.		Memory area of PLC (1 ... 32)
WN	MW addr.		<WN> = starting address of data in memory area (1 ... x) with x $\leq$ (end address im memory area - 800).
VI2	MW addr.	2	Consecutive, for organization information
RDY	Bit addr.		Ready signal: RDY = 1 $\triangleq$ idle state
AF	Bit addr.		AF = 1: error
WAF	MW addr.		Error status

NWM New-Value Signal (SFB116)

Formal operand	**Identi-fier**	**Meaning**
NWM		Operation (call)
EF	Bit addr.	Enable
QS	Bit addr.	Acknowledgement of new-value signals
RV	Bit addr.	Acknowledgement of events off
ZW	MW addr.	<ZW> = current states of the events
ZQ	MW addr.	<ZQ> = new-value events
ZA	MW addr.	<ZA> = non-reset events acknowledged with QS
ZV	MW addr.	<ZV> = events off, acknowledged with QS
ZM	MW addr.	<ZM> = event signals displayed
AM	Bit addr.	Signal for new-value event

O-REG Organization Block Closed-loop Control (SFB390)

Formal operand	**Identi-fier**	**Meaning**
O-REG		Operation (call)
CRT	Bit addr.	”1” = Activate organization block entered in OB
ZVT	ZVT?	Data structure time management table, see manual ”A350/A500,Regeln mit Dolog AKF, Benutzerhandbuch”[9)]
OB	OB?	Organization block in which the closed-loop control is configured (e.g. OB2)
AF	Bit addr.	AF = 1: error
WAF	Word addr.	Error code

9) in german language

PBM Pulse Width Modulation (SFB320)

Formal operand	**Identifier**	**Meaning**
PBM		Operation (call)
STAT	Word addr.	Status control loop: -2 = Halt, -1 = Reset, 0 = Initial, 1 = Running
RST	Bit addr.	Operating mode Reset (”1” = Reset)
PARA	PPBM?	Data structure parameters (see below)
X	FWord addr.	Input quantity
AUF	Bit addr.	Output AUF
ZU	Bit addr.	Output ZU
AF	Bit addr.	AF = 1: error
WAF	Word addr.	Error code
VI	VIA?	Data structure organization information

PPBM

Element	**Element type**	**Symbol suggestion**	**Meaning**
PPBMn.x			Data structure parameters of PBM, n = 100
PPBMn.1	Float word	t-period	Period length in sec
PPBMn.2	Float word	tmin	Minimum time for actuating pulse
PPBMn.3	Float word	tmax	Maximum time for actuating pulse
PPBMn.4	Float word	t-pause	Pause time in sec
PPBMn.5	Float word	t-brems	Brake time in sec
PPBMn.6	Float word	gob-pos	Upper limit for X positive
PPBMn.7	Float word	gob-neg	Lower limit for X negative

PDM Pulse Duration Modulation (SFB325)

Formal operand	Identi- fier	Meaning
PDM		Operation (call)
STAT	Word addr.	Status control loop: -2 = Halt, -1 = Reset, 0 = In-
itial,		
		1 = Running
RST	Bit addr.	Operating mode Reset (”1” = Reset)
PARA	PPDM?	Data structure parameters (see below)
X	FWord addr.	Input quantity
AUF	Bit addr.	Output AUF
ZU	Bit addr.	Output ZU
AF	Bit addr.	AF = 1: error
WAF	Word addr.	Error code
VI	VIA?	Data structure organization information

PPDM

Element	Element type	Symbol suggestion	Meaning
PPDMn.x			Data structure parameters of PDMO, n = 50
PPDMn.1	Float word	t-ein	Pulse length in sec
PPDMn.2	Float word	t-pause	Pause time in sec
PPDMn.3	Float word	t-brems	Brake time in sec
PPDMn.4	Float word	gob-pos	Upper limit for X positive
PPDMn.5	Float word	tmin-auf	Minimum cycle time for AUF (if x = gob-pos)
PPDMn.6	Float word	gub-pos	Lower limit for X positive
PPDMn.7	Float word	tmax-auf	Maximum cycle time for AUF (if x = gub-pos)
PPDMn.8	Float word	gob-neg	Upper limit for X negative
PPDMn.9	Float word	tmin-zu	Minimum cycle time for ZU (if x = -gob-neg)
PPDMn.10	Float word	gub-neg	Lower limit for X negative
PPDMn.11	Float word	tmax-zu	Maximum cycle time for ZU (if x = -gub-neg)

PDT1 PD-Element with Smoothing (SFB340)

Formal operand	Identi- fier	Number	Meaning
PDT1			Operation (call)
STAT	Word addr.		Status control loop: -2 = Halt, -1 = Reset, 0 = In- itial, 1 = Running
RST	Bit addr.		Reset operating mode (”1” = Reset)
HALT	Bit addr.		Halt operating mode (”1” = Halt)
PARA	PPDT?		Data structure parameters (see below)
X	FWord addr.		Input
YRST	FWord addr.		Reset value output
Y	FWord addr.		Output
AF	Bit addr.		AF = 1: error
WAF	Word addr.		Error code
VI	VIA?	16 Byte	Data structure organization information

PPDT

Element	Element type	Symbol suggestion	Meaning
PPDTn			Data structure parameters of PDT1, n = 50
PPDTn.1	Float word	GK	Gain
PPDTn.2	Float word	TD	Derivative action time constant
PPDTn.3	Float word	T1	Smoothing time constant

PI PI-Controller (SFB308)

Formal operand	Identi-fier	Number	Meaning
PI			Operation (call)
STAT	Word addr.		Status control loop: -2 = Halt, -1 = Reset, 0 = Initial, 1 = Running
HAND	Bit addr.		Manual operating mode (”1” = Hand)
HALT	Bit addr.		Halt operating mode (”1” = Halt)
PARA	PPI?		Data structure parameters (see below)
W	FWord addr.		Setpoint value input
X	FWord addr.		Actual value input
YHND	FWord addr.		Manual manipulated variable
XD	FWord addr.		Output system deviation
Y	FWord addr.		Manipulated variable
MELD	MPID?		Data structure signals (see below)
AF	Bit addr.		AF = 1: error
WAF	Word addr.		Error code
VI	VIA?	16 Bytes	Data structure organization information

PPI

Element	Element type	Symbol suggestion	Meaning
PPIn			Data structure parameters of PI, n = 99
PPIn.1	Float word	Kp	Proportional rate (gain)
PPIn.2	Float word	Tn	Reset time
PPIn.3	Float word	YAO	Upper controller output limit
PPIn.4	Float word	YAU	Lwoer controller output limit

MPID

Element	Element type	Symbol suggestion	Meaning
MPIDn			Data structure signals, n = 227
MPIDn.1	Bit	aYAO	”1” = signal: Y reached upper controller output limit
MPIDn.2	Bit	AYAU	”1” = signal: Y reached lower controller output limit

PID PID-Controller (SFB302)

Formal operand	Identifier	Number	Meaning
PID			Operation (call)
STAT	Word addr.		Status control loop: -2 = Halt, -1 = Reset, 0 = Initial, 1 = Running
BT	BPID		Data structure operating modes (see below)
PARA	PPID		Data structure parameters (see below)
W	FWord addr.		Setpoint value input
X	FWord addr.		Actual value input
YHND	FWord addr.		Manual manipulated variable
Z	FWord addr.		Disturbance input
XD	FWord addr.		Output system deviation
Y	FWord addr.		Manipulated variable
MELD	MPID?		Data structure signals
AF	Bit addr.		AF = 1: error
WAF	Word addr.		Error code
VI	VIB?	32 Byte	Data structure organization information

BPID

Element	Element type	Symbol suggestion	Meaning
BPIDn			Data structure operating modes of PID, n = 99
BPIDn1.	Bit	Hand	Control input for manual operating mode ("1" = Hand)
BPIDn.2	Bit	Halt	Control input for Halt operating mode ("1" = Halt)
BPIDn.3	Bit	P-Ein	"1" = P-part on
BPIDn.4	Bit	I-Ein	"1" = I-part on
BPIDn.5	Bit	D-Ein	"1" = D-part on
BPIDn.6	Bit	EDUM	"1" = D-part to controlled variable, "0" = D-part to system deviation

PPID

Element	Element type	Symbol suggestion	Meaning
PPIDn			Data structure parameters of PID, n = 99
PPIDn.1	Float word	KP	Proportional rate (gain)
PPIDn.2	Float word	Tn	Reset time
PPIDn.3	Float word	Tv	Lead time
PPIDn.4	Float word	OH	Upper controller output limit
PPIDn.5	Float word	UH	Lower controller output limit

MPID

Element	Element type	Symbol suggestion	Meaning
MPIDn			Data structure signals, n = 227
MPIDn.1	Bit	AYAO	”1” = signal: Y reached upper controller output limit
MPIDn.2	Bit	AYAU	”1” = signal: Y reached lower controller output limit

Formal operand	Identi-fier	Number	Meaning
PIDP			Operation (call)
STAT	FWord addr.		Status control loop: -2 = Halt, -1 = Reset, 0 = Initial, 1 = Running
BT	BIDP?		Data structure operating modes (see below)
PARA	PIDP?		Data structure parameters (see below)
W	FWord addr.		Setpoint value input
X	FWord addr.		Actual value input
YHND	FWord addr.		Manual manipulated variable
Z	FWord addr.		Disturbance
XD	FWord addr.		Output system deviation
Y	FWord addr.		Manipulated variable
MELD	MPID?		Data structure signals
AF	Bit addr.		AF = 1: error
WAF	Word addr.		Error code
VI	VIB?	32 Byte	Data structure organization information

BIDP

Element	Element type	Symbol suggestion	Meaning
BIDPn			Data structure operating modes of PIDP, n = 30
BIDPn.1	Bit	Hand	Control input for manual operating mode (”1” = Hand)
BIDPn.2	Bit	Halt	Control input for Halt operating mode (”1” = Halt)
BIDPn.3	Bit	EDUM	”1” = D-part to controlled variable, ”0” = D-part to system deviation

PIDP

Element	Element type	Symbol suggestion	Meaning
PIDPn			Data structure parameters of PIDP, n = 30
PIDPn.1	Float word	KP	Proportional rate (gain)
PIDPn.2	Float word	KI	Integral-action factor
PIDPn.3	Float word	KD	Derivative-action factor
PIDPn.4	Float word	YAO	Upper controller output limit
PIDPn.5	Float word	YAU	Lower controller output limit

MPID

Element	Element type	Symbol suggestion	Meaning
MPIDn			Data structure signals, n = 227
MPIDn.1	Bit	AYAO	”1” = signal: Y reached upper controller output limit
MPIDn.2	Bit	AYAU	”1” = signal: Y reached lower controller output limit

PLA — Program Transmission PLC → POS (SFB242)

Formal operand	Identifier	Meaning	
PLA		Operation (call)	
EF	Bit addr.	Enable	
SP	MW addr.	Slot reference POS	
WB	MW addr.	Memory area	PLC
WA	MW addr.	Offset	
VI2	MW addr.	Internal organization information	
RDY	Bit addr.	Ready signal	

POLY — Polynomial Computation (SFB295)

Formal operand	Identifier	Number	Meaning
POLY			Operation (call)
EF	Bit addr.		EF = 1: Enable
WEN	MW addr.	1+10MG	<WEN> = Polynomial degree (0 ... 10), subsequent 10 MF addresses contain the factors (a_o ... a_{10})
GE	MF addr.		<GE> = Base of the power
GA	MF addr.		<GA> = Result of the polynomial computation
AF	Bit addr.		AF = 1: error
WAF	MW addr.		Error code

POS Position Transmission (SFB248)

Formal operand	Identi-fier	Meaning
POS		Operation (call)
EF	Bit addr.	Enable
BT	Bit addr.	Job
SP	MW addr.	Slot reference POS
EB	Bit addr.	Acceleration
EE	Bit addr.	Rapid traverse
EK	Bit addr.	Incremental dimension
DWV	MD addr.	Feed
DWW	MD addr.	Position value
DWZ	MD addr.	Additional function
VI1	MW addr.	Organization
RDY	Bit addr.	Ready signal
WAF	MW addr.	Error code (<MW> 768/1024/3072)

POT Exponentiation (SFB296)

Formal operand	Identi-fier	Number	Meaning
POT			Operation (call)
EF	Bit addr.		EF = 1: Enable
GE	MF addr.		<GE> = Input quantity base
GEX	MF addr.		<GEX> = Input quantity exponent
GA	MF addr.		<GA> = Output quantity power
AV	Bit addr.		Sign of power
AF	Bit addr.		AF = 1: error
WAF	MW addr.		<WAF> = Error code

POV Position Presetting (SFB249)

Formal operand	Identi-fier	Meaning
POV		Operation (call)
EF	Bit addr.	Enable
BT	Bit addr.	Job
SP	MW addr.	Slot reference POS
ABS	Bit addr.	Absolute value definition
DWP	MD addr.	Position
VI1	MW addr.	Internal organization information
RDY	Bit addr.	Ready signal
WAF	MW addr.	Error code

PRT Log Element (SFB199)

Formal operand	Identi-fier	Meaning
PRT		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Reset
BK	Bit addr.	1st initial signal
BL	Bit addr.	Last initial signal
SSB	MW addr.	<SSB> = memory area of signal file
SNR	MW addr.	<SNR> = file number of signal file
GNR	MW addr.	<GNR> = device number
PUA	MW addr.	Start of buffer (in signal memory)
PUE	MW addr.	End of buffer (in signal memory)
FN	MW addr.	Error code
AB	Bit addr.	Automatic output limit; 1 = impending buffer overflow
RDY	Bit addr.	Ready signal; 0 = output; 1 else

PT1 Time-Delay Element 1st Order (SFB330)

Formal operand	Identi- fier	Meaning-
PT1		Operation (call)
STAT	Word addr.	Status control loop: -2 = Halt, -1 = Reset, 0 = Initial, 1 = Running
RST	Bit addr.	Reset operating mode (”1” = Reset)
HALT	Bit addr.	Halt operating mode (”1” = Halt)
PARA	PPT?	Data structure parameters (see below)
X	FWord addr.	Input
YRST	FWord addr.	Reset value output
Y	FWord addr.	Output
AF	Bit addr.	AF = 1: error
WAF	Word addr.	Error code
VI	VIA?	Data structure organization information

PPT

Element	Element type	Symbol suggestion	Meaning
PPTn.x			Data structure parameters of PT1, n = 99
PPTn.1	Float word	GK	Gain
PPTn.2	Float word	T1	Delay time constant

PT2 Time-Delay Element 2nd Order (SFB331)

Formal operand	Identi-fier	Number	Meaning
PT2			Operation (call)
STAT	Word addr.		Status control loop: -2 = Halt, -1 = Reset, 0 = Initial, 1 = Running
RST	Bit addr.		Reset operating mode (”1” = Reset)
HALT	Bit addr.		Halt operating mode (”1” = Halt)
PARA	PPTT?		Data structure parameters (see below)
X	FWord addr.		Input
YRST	FWord addr.		Reset value output
Y	FWord addr.		Output
AF	Bit addr.		AF = 1: error
WAF	Word addr.		Error code
VI	VIA?	16 Bytes	Data structure organization information

PPTT

Element	Element type	Symbol suggestion	Meaning
PPTTn.x			Data structure parameters of PT2, n = 50
PPTTn.1	Float word	GK	Gain
PPTTn.2	Float word	d	Damping
PPTTn.3	Float word	omega	Natural frequency

REF Accept Reference Point (SFB244)

Formal operand	Identi-fier	Number	Meaning
REF			Operation (call)
EF	Bit addr.		Enable
SP	MW addr.		Slot reference POS
EP	Bit addr.		Direction <EP> = 0: minus, <EP> = 1: plus
DWV	MD addr.		Feed
RDY	Bit addr.		Ready signal
WAF	MW addr.		Error code (<MW> 256/512/2816)

Formal operand	Identi-fier	Number	Meaning
SA03			Operation (call)
EF	Bit addr.	1	Enable
ER	Bit addr.	1	Reset block
ES	Bit addr.	1	Write access enable for RS 232 C interface of SAA/SAI 103, ES = 1; data modification with the RS 232 C interface possible
SP	MW addr.	1	<SP> = Slot reference (slot); 2 ... 160
UF	MD addr.	3	Scaling factor to convert IW
UNP	MD addr.	3	Zero point shift to convert IW
VI2	MW addr.	2	Organization information
AAK	Bit addr.	3	AAK = 1: axis activated
MEA	MW addr.	3	Currently processed endstop; <MEA> = 1 ... 5
IW	MD addr.	3	Actual value (position)
VA1E	Bit addr.	3	Value of prestop 1; negation of outputs (NVAA) is ignored
VA2E	Bit addr.	3	Value of prestop 2; negation of outputs (NVAA) is ignored
VA3E	Bit addr.	3	Value of prestop 1; negation of outputs (NVAA) is ignored
VA4E	Bit addr.	3	Value of prestop 2; negation of outputs (NVAA) is ignored
PRE	Bit addr.	3	Value of plus direction; negation of output (NRA) is ignored
MRE	Bit addr.	3	Value of minus direction; negation of output (NRA) is ignored
RIUE	Bit addr.	3	Direction specification RIUE = 0 $\rightarrow$ positive direction RIUE = 1 $\rightarrow$ negative direction
EAE	Bit addr.	3	Value of endstop
IWRE	Bit addr.	3	only SAI; IWRE = 1: actual value reset
VEAE	Bit addr.	3	VEAE = 1: pre-endstop
REFE	Bit addr.	3	only SAI; REFE = 1: reference point reached
EE	Bit addr.	8	Value of process inputs 1 ... 8 of SAA/SAI
AP1	Bit addr.	1	AP1 = 1: Setpoint field modification
AP2	Bit addr.	1	AP2 = 1: Initialization field modification
AF	Bit addr.	1	AF = 1: error
WAF	MW addr.	1	Error code

Formal operand	Identi- fier	Number	Meaning
SA03E			Operation (call)
EF	Bit addr.		Enable
ER	Bit addr.	1	Reset block
BA	MW addr.	1	Block selection <BA> = 1; SAI 103 <BA> = 2; SAA 103
UEBR	Bit addr.	1	Transmission direction: UEBR = 0; SAA/SAI → controller UEBR = 1; controller → SAA/SAI
ES	Bit addr.	1	Write access enable for the RS 232 C interface of the SAA/SAI, ES = 1: data modification possible with the RS 232 C interface Transmission direction only controller → SAA/SAI
ESR	Bit addr.	1	Recovery of write access authority on the SAA/SAI for the controller if 0 → 1 edge Transmission direction only controller → SAA/SAI
EQ	Bit addr.	1	Short circuit acknowledgement if 0 → 1 edge Transmission direction only controller → SAA/SAI
VI2	MW addr.	2	Organization information
SP	MW addr.	1	<SP> = Slot reference (slot); 2 ... 160
UF	MD addr.	3	Scaling factor
UNP	MD addr.	3	Zero point shift
EFB1	Bit addr.	1	Block enable 1 (setpoint value field)
FA	Bit addr.	3	Drive command; start axis if FA = 1, taking drive command input into consideration
EA1	MD addr.	15	Endstop values (one endstop value per switching axis):
BRES	Bit addr.	3	Reset axis; BRES = 1: preset standard values for the relevant axis
EFB2	Bit addr.	1	Block enable 2 (initialization data)
AK	Bit addr.	3	Activate axis; only the activated axes of the SAA/SAI are processed AK = 1: axis activated
MA	MW addr.	3	Resolution of sensor, i.e. number of increments per revolution; <MA> = 1 ... 30 000
MU	MW addr.	3	for SAI: pulse evaluation mode; <MU> = 1: 1 logic <MU> = 2: 1/2 logic; <MU> = 4: 1/4 logic for SAA: number of revolutions of sensor; <MU> = 1 ... 30 000
TZ	MW addr.	3	only SAA; number of clock pulses of sensor; <TZ> = 0: no. of clock pulses is determined autom. <TZ> = 1 ... 30: Direct def. of no. of clock pulses

Continued on next page

Formal- operand	**Identi- fier**	**Number**	**Meaning**
SA03E continued			
BETR	MW addr.	3	Operating mode; 1 = only SAI; search for reference linear operat. 2 = linear operation 3 = rotary axis operation 4 = only SAI; search for reference rotary axis op.
VA1	MD addr.	12	Prestop values (one prestop value each for axis 1 and 2)
GO	MD addr.	3	only SAA 103; sensor offset
IEA	Bit addr.	3	IEA = 1; enable endstop input (input 8)
RDY	Bit addr.	1	Ready signal; RDY = 0 during data transmission between PLC and SAA/SAI
AF	Bit addr.	1	Error bit; if an error occurred, AF = 1
WAF	W addr.	1	Error code; <WAF> = error nummer

SA03I Parametrization for SAA/SAI 103 (SFB15)

Formal operand	**Identi- fier**	**Number**	**Meaning**
SA03I			Operation (call)
EF	Bit addr.		Enable
ER	Bit addr.	1	Reset block
BA	MW addr.	1	Module selection <BA> = 1: SAI 103 <BA> = 2: SAA 103
UEBR	Bit addr.	1	Transmission direction; UEBR = 0: SAA/SAI → controller UEBR = 1: controller → SAA/SAI
ES	Bit addr.	1	Write access enable for the RS 232 C interface of the SAA/SAI, ES = 1: data modification possible with the RS 232 C interface Transmission direction only controller → SAA/SAI
ESR	Bit addr.	1	Recovery of write access authority on the SAA/SAI for the controller if 0 → 1 edge Transmission direction only controller → SAA/SAI
EQ	Bit addr.	1	Short circuit acknowledgement if 0 → 1 edge Transmission direction only controller → SAA/SAI 103
VI2	MW addr.	2	Organization information
SP	MW addr.	1	<SP> = Slot reference (slot); 2 ... 160
UF	MD addr.	3	Scaling factor

Continued on next page

Formal operand	Identi-fier	Number	Meaning
SA03I continued			
UNP	MD addr.	3	Zero point shift
EFB1	Bit addr.	1	Block enable 1 (setpoint value field)
FA	Bit addr.	3	Drive command; start axis if FA = 1, taking UVO and IS into consideration
EA1	MD addr.	15	Endstop values (5 endstop values per switching axis): 3 x EA1, 3 x EA2, ... , 3 x EA5
LEA	MW addr.	3	Last endstop to be approached; <LEA> = 1 ... 5
BEA	Bit addr.	3	Pre-endstop; BEA = 1: Activate prestop and endstop
IWR	Bit addr.	3	only SAI; reset actual value; only allowed for operation modes 1 and 4; IWR = 1: actual value is set to 0
BRES	Bit addr.	3	Reset axis; BRES = 1: preset standard values for the relevant axis
EFB2	Bit addr.	1	Block enable 2 (initialization data
AK	Bit addr.	3	Activate axis; only the activated axes of the SAA/SAI are processed AK = 1: axis activated
PK	Bit addr.	3	Parametrize axis; the initialization data are only accepted for the axes for which PK = 1.
EBUA	Bit addr.	3	Switch-off behavior of the SAA/SAI after expiration of the PLC monitoring time; EBUA = 1: activate prestop and endstop EBUA = 0: retain function of the SAA/SAI
TUE	MW addr.	1	a) PLC monitoring time 1 ... 255 x 100 msec; after expiration of the monitoring time, the reaction defined in EBUA occurs b) Extension of the maximum permitted program scan time of the PLC beyond 100 msec by 1 ... 255 x 10 msec
MA	MW addr.	3	Resolution of the sensor, i.e. number of increments per revolution; <MA> = 1 ... 30 000
MU	MW addr.	3	for SAI: pulse evaluation mode; <MU> = 1: full logic; <MU> = 2: 1/2 logic; <MU> = 4: 1/4 logic for SAA: number of revolutions of sensor; <MU> = 1 ... 30 000
TZ	MW addr.	3	only SAA; number of clock pulses of sensor; <TZ> = 0: no. of clock pulses is determined autom. <TZ> = 1 ... 30 direct def. of the no. of clock pulses
GC	Bit addr.	3	only SAA; sensor code; GC = 0: gray code GC = 1: binary code

Continued on next page

Formal operand	Identi-fier	Number	Meaning
SA03I continued			
GFU	Bit addr.	3	Sensor error suppression: GFU = 0: no suppression GFU = 1: for SAA suppress double read for SAI suppress phase error
BETR	MW addr.	3	Operating mode; 1 = only SAI; search for reference linear operation 2 = linear operation 3 = rotary axis operation 4 = only SAI; search for reference rotary axis op.
AVA	Bit addr.	3	Type of prestop; AVA = 0: parallel prestops, AVA = 1: serial prestops
EAF	Bit addr.	3	Floating endstop; EAF = 0: normal operation EAF = 1: no halt at the individual endstop values
WZ	MW addr.	3	Waiting time between endstop and hot restart <WZ> = 0 ... 255 x approx. 20 msec + approx. 20 msec (approx. 20 msec = 5 times the scan time of the SAA/SAI)
VA1	MD addr.	12	Prestop values (4 prestop values per switching axis)
STBP	MD addr.	6	In-position window; first plus direction for all switching axes, then minus direction
ESO	MD addr.	6	Endstop values; first upper then lower values
GO	MD addr.	3	only SAA 103; sensor offset
NE	Bit addr.	8	Negate process input; NE = 1: negate input
UVO	Bit addr.	3	Type of logic operation for the start inputs defined with IS; UVO = 0: And operation for inputs, UVO = 1: Or operation for inputs
IS	MW addr.	3	Input selection for the drive command (FA) of the axis. Maximum 3 inputs per axis
IER	MW addr.	3	Input selection for reset axis (BRES), pre-endstop (BEA) and reference point signal (only for SAI), for SAA: 1 input for BRES per axis, maximum 3 inputs for BEA per axis, for SAI: 1 input for BRES per axis 1 input for the reference point per axis, maximum 2 inputs for BEA per axis
ISMP	MW addr.	3	Input selection for manual control of the outputs MRE and PRE and selection of the high-speed step (the prestop outputs VA1 ... VA4 are set accordingly)
IMPE	MW addr.	3	Input selection for the endstop of the plus and minus directions

Continued on next page

Formal operand	**Identi-fier**	**Number**	**Meaning**
SA03I continued			
VAA	MW addr.	3	Output selection for the prestops (VA1E ... VA4E)
NVAA	MW addr.	3	Negation of the outputs defined in VAA
RA	MW addr.	3	Output selection for the drive command, the end-stop (EAE), plus direction (PRE) and minus direction (MRE)
NRA	MW addr.	3	Negation of the outputs defined in RA
RDY	Bit addr.	1	Ready signal; during the data transmission between PLC and SAA/SAI, RDY = 0
AF	Bit addr.	1	Error bit; if an error occurred, AF = 1
WAF	W addr.	1	Error code; <WAF> = error nummer

SAB Output Station for SRB (SFB236)

Formal operand	**Identi-fier**	**Number**	**Meaning**
SAB			Operation (call)
VI	MW addr.	4	Starting address internal organization information
EP	MW addr.		<EP> = Position in register : 0 ... <RL>-1
ER	Bit addr.		ER = 1: Delete position (register location)
A	Bit addr.		Output (target in signal memory)

SAS Output Station Serial for SRW (SFB241)

Formal operand	**Identi-fier**	**Number**	**Meaning**
SAS			Operation (call)
VI	MW addr.	4	Starting address internal organization information
EP	MW addr.		<EP> = Position in register : 0 ... <RL> - 1
ER	Bit addr.		ER = 1: Delete position (register word location)
AK	Bit addr.		Output (lowest address target area in signal memory)

SAW Output Station Word for SRW (SFB240)

Formal operand	Identi-fier	Number	Meaning
SAW			Operation (call)
VI	MW addr.	4	Starting address internal organization information
EP	MW addr.		<EP> = Position in register: 0 ... <RL>-1
ER	Bit addr.		ER = 1: Delete position (register word location)
WA	MW addr.		Output (target word in signal memory)

SB Shift Bits in Marker Area (SFB122)

Formal operand	Identi-fier	Number	Meaning
SB			Operation (call)
RR	Bit addr.		Reset
EF	Bit addr.		Enable
ST	Bit addr.		Shift clock pulse
K	Bit addr.		Length code: K = 0: 8 bits, K = 1: 16 bits
E	Bit addr.		Shift register - input
FK	Bit addr.		Previous state of ST
A	Bit addr.		Outputs (1st address of 8 or 16 bits)

SBVE Preset Memory Area (SFB264)

Formal operand	Identi-fier	Number	Meaning
SBVE			Operation (call)
EF	Bit addr.		Enable
WBU	MW addr.		<WBU> = memory area to be standardized
WZ	MW addr.		<WZ> = starting address
WN	MW addr.		<WN> = amount of data
WE	MW addr.		<WE> = value with which to standardize

SEB Input Station for SRB (SFB235)

Formal operand	Identi- fier	Number	Meaning
SEB			Operation (call)
VI	MW addr.	4	Starting address internal organization information
EP	MW addr.		<EP> = Position in register: 0 ... <RL>-1
EU	Bit addr.		EU = 1: Read in E to <EP> EU = 0: do not read in
E	Bit addr.		Input (source in signal memory)
AE	Bit addr.		AE = 1: position empty, read in is possible

SEIG Read in Interface, device-dependent (SFB262)

Formal operand	Identi- fier	Meaning
SEIG		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Reset
GNR	MW addr.	<GNR> = Device number
FN	MW addr.	Error number

SEIN Read in Interface (SFB198)

Formal operand	Identi- fier	Meaning
SEIN		Operation (call)
EF	Bit addr.	Enable

SES Input Station Serial for SRW (SFB239)

Formal operand	Identi-fier	Number	Meaning
SES			Operation (call)
VI	MW addr.	4	Starting address internal organization information
EP	MW addr.		<EP> = Position in register: 0 .. <RL>-1
EU	Bit addr.		EU = 1: Read in <EK ...> → <EP>, EU = 0: do not read in
EK	Bit addr.		Input lowest bit address of <RB> bits, (source area signal memory)
AE	Bit addr.		AE = 1: Position empty, read in is possible

SEW Input Station Word for SRW (SFB238)

Formal operand	Identi-fier	Number	Meaning
SEW			Operation (call)
VI	MW addr.	4	Starting address internal organization information
EP	MW addr.		<EP> = Position in register: 0 ... <RL>-1
EU	Bit addr.		EU = 1: Read in <WE> → <EP>, EU = 0: do not read in
WE	MW addr.		Input (source word in signal memory)
AE	Bit addr.		AE = 1: Position empty, read in is possible

SFW Shift Bit in Word (SFB123)

Formal operand	Identi-fier	Meaning
SFW		Operation (call)
EF	Bit addr.	Enable
LR	Bit addr.	Shift direction: LR = 0: right, LR = 1: left
RS	Bit addr.	1 = End-around shift right
DW	MW addr.	<DW> = Word to be shifted
DA	MW addr.	<DA> = Number of shift clock pulses (1 - 16)
AB	Bit addr.	Value of last bit shifted out

SHF Shift Field Word Area (SFB125)

Formal operand	Identi-fier	Meaning
SHF		Operation (call)
EF	Bit addr.	Enable
EK	Bit addr.	Shift direction, 0 = right, 1 = left shift
WN	MW addr.	<WN> = Number of shifts (0 - 16)
WEN	MW addr.	<WEN>= Information to be shifted up
WK	MW addr.	<WK> = Address of first word to be shifted
WL	MW addr.	<WL> = Address of last word to be shifted

SHW Shift Word (SFB124)

Formal operand	Identi-fier	Meaning
SHW		Operation (call)
EF	Bit addr.	Enable
EK	Bit addr.	Shift direction: <EK>=0: right; <EK>=1: left
WN	MW addr.	<WN> = Number of shifts (1 - 15)
WE	MW addr.	<WE> = Word to be shifted
WEN	MW addr.	<WEN> = Information to be shifted up
WA	MW addr.	<WA> = Output word

SIN Sine Function (SFB276)

Formal operand	Identi-fier	Meaning
SIN		Operation (call)
EF	Bit addr.	Enable
GE	MF addr.	<GE> = Input quantity in radian measure
GA	MF addr.	<GA> = Output quantity
AF	Bit addr.	AF=1: error
WAF	MW addr.	Error code

SPM Peak Value Signalling (SFB184)

Formal operand	Identi-fier	Meaning
SPM		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Reset
MW	MW addr.	<MW> = Current measured value
WG	MW addr.	<WG> = Largest measured value so far

SRB Shift Register for Bit Processing (SFB234)

Formal operand	Identi-fier	Number	Meaning
SRB			Operation (call)
VI	MW addr.	4	Internal organization information
RL	MW addr.		<RL> = Register length: 1...
RR	Bit addr.		Total delete (register locations, pointer position)
ST	Bit addr.		Shift clock pulse
SI	Bit addr.		Shift direction
EP	MW addr.		<EP> = Observer position: 0 ... <RL> - 1
EU	Bit addr.		EU = 1: Overwrite location with value of EI EU = 0: Location contents copied to EI
RA	Bit addr.		Start of register direct, lowest bit field address
EI	Bit addr.		Information from observer position

SRW Shift Register for Word Processing (SFB237)

Formal operand	Identi-fier	Number	Meaning
SRW			Operation (call)
VI	MW addr.	4	Internal organization information
RA	MW addr.		Start of register direct, lowest word field address
RL	MW addr.		<RL> = Register length: 1 ...
RB	MW addr.		<RB> = Register width: 1 ... 8
RR	Bit addr.		Total delete (register locations, pointer position)
ST	Bit addr.		Shift clock pulse
SI	Bit addr.		Shift direction
EP	MW addr.		<EP> = Observer position
EU	Bit addr.		EU = 1: Overwrite location with value of EI, EU = 0: Location contents copied to EI
EI	MW addr.		Information from observer position

STOE Signal Disturbance (SFB360)

Formal operand	Identi-fier	Meaning
STOE		Operation (call)
PARA	PSTO	Data structure parameters (see below)
X	FWord addr.	Actual value input
STOE	Bit addr.	Signal input, X is disturbed (=1)
Y	FWord addr.	Output
OK	Bit addr.	”1” = Y is OK
INIT	Bit addr.	”1” = 1st value of Y after disturbance
STA	Bit addr.	Previous value disturbance

PSTO

Element	Element type	Symbol suggestion	Meaning
PSTOn.x			Data structure parameters of STOE, n = 50
PSTOn.1	Float word	OG	Upper limit for X (value larger: disturbance)
PSTOn.2	Float word	UG	Lower limit for X (value smaller: disturbance)

SUE Subtraction Word (SFB157)

Formal operand	Identi-fier	Meaning
SUE		Operation (call)
WE2	MW addr.	<WE2> = Subtrahend
WE1	MW addr.	<WE1> = Minuend
WA	MW addr.	<WA> = Result
AF	Bit addr.	AF = 1: error

SWM Threshold Value Comparison (SFB183)

Formal operand	Identi-fier	Meaning	
SWM		Operation (call)	
EF	Bit addr.	Enable	
ER	Bit addr.	Reset	
SW	MW addr.	<SW> =	threshold value to be specified
MW	MW addr.	<MW> =	current measured value
WA	MW addr.	<WA> =	old measured value
WN	MW addr.	<WN> =	computed difference <MW> - <WA>
AS	Bit addr.	AS = 1: <SW> violated; <WN> ≥ <SW>	
AF	Bit addr.	Error bit has no meaning	

TAN Tangent Function (SFB278)

Formal operand	Identi-fier	Meaning	
TAN		Operation (call)	
EF	Bit addr.	Enable	
GE	MF addr.	<GE> =	Input quantity in radian measure
GA	MF addr.	<GA> =	Output quantity
AF	Bit addr.	AF=1: error	
WAF	MW addr.	Error code	

TEA Text Output (SFB203)

Formal operand	Identi-fier	Meaning
TEA		Operation (call)
EF	Bit addr.	Enable
BT	Bit addr.	Commission: Output if 0/1-edge
GNR	MW addr.	<GNR> = device number
TBO	MW addr.	<TBO> = text block upper limit
RDY	Bit addr.	Ready signal; 0 = output; 1 else
MRY	MW addr.	User memory for TEA
FN	MW addr.	Error code

TEE Text Input (SFB200)

Formal operand	Identi-fier	Meaning
TEE		Operation (call)
EF	Bit addr.	Enable
BT	Bit addr.	Commission: Read in if 0/1-edge
GNR	MW addr.	<GNR> = Device number
ZA	MW addr.	<ZA> = Maximum number of characters
RDY	Bit addr.	Ready signal; 0 = output; 1 else
TBO	MW addr.	<TBO> = Text block upper limit
TBU	MW addr.	<TBU> = Text block lower limit
MRY	MW addr.	User memory for TEE
FN	MW addr.	Error code

TEEI Text Input, interrupt-control (SFB202)

Formal operand	Identi-fier	Meaning
TEEI		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Reset
GNR	MW addr.	<GNR> = Device number
ZA	MW addr.	<ZA> = Maximum number of characters
RDY	Bit addr.	Ready signal; 0-1-0 after reading in/abort; 0 else
TBO	MW addr.	<TBO> = Text block upper limit
TBU	MW addr.	<TBU> = Text block lower limit
AF	Bit addr.	AF = 1: error

TEEZ Text Input with Time Limit (SFB201)

Formal operand	Identi-fier	Meaning
TEEZ		Operation (call)
EF	Bit addr.	Enable
BT	Bit addr.	Commission: Read in if 0/1-edge
GNR	MW addr.	<GNR> = Device number
ZA	MW addr.	<ZA> = Maximum number of characters
WT	MW addr.	<WT> = Wait time 1 ... 255 sec; <WT> > 255: Wait time infinite
RDY	Bit addr.	Ready signal; 0 = reading in; 1 else
TBO	MW addr.	<TBO> = Text block upper limit
TBU	MW addr.	<TBU> = Text block lower limit
MRY	MW addr.	User memory for TEEZ
FN	MW addr.	Error code

TEV Compare Text (SFB204)

Formal operand	Identi-fier	Meaning
TEV		Operation (call)
EF	Bit addr.	Enable
BT	Bit addr.	Commission: Compare for 0/1-edge
TBO	MW addr.	<TBO> = Text block upper limit
TSB	MW addr.	<TSB> = Memory area of comparison texts
TDN	MW addr.	<TDN> = File number of comparison texts
TZN	MW addr.	<TZN> = 1st line of file to be compared
AZ	Bit addr.	Previous state of signal BT
BK	Bit addr.	Bit for 1st text line to be compared
BL	Bit addr.	Bit for last text line to be compared
RZN	MW addr.	<RZN> = Same line number, relative to <TZN>; <RZN> = 0: unequal
AF	Bit addr.	AF = 1: error

TKA Output Comnmunications (SFB255)

Formal operand	Identi-fier	Meaning
TKA		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Reset
BT	Bit addr.	Commission: Send if 0/1-edge
CHK	Bit addr.	Checksum byte; 1 = append
GNR	MW addr.	<GNR> = Device number
TBO	MW addr.	<TBO> = Upper limit text block address
EBO	MW addr.	<EBO> = Text block address for echo reception
ZE	MW addr.	<ZE> = Number of characters echo
WT	MW addr.	<WT> = Wait time; unit = 10 msec
RDY	Bit addr.	Ready signal; 0 = when sending; 1 else
SBO	MW addr.	<SBO> = Text block address for call; <SBO>-1 = Number
MRY	MW addr.	Internal memory
FN	MW addr.	Error code

TKE Input Communications (SFB254)

Formal operand	Identi-fier	Meaning
TKE		Operation (call)
EF	Bit addr.	Enable
ER	Bit addr.	Reset
BT	Bit addr.	Commission: Send if 0/1-edge
CHK	Bit addr.	Checksum byte; 1 = append
GNR	MW addr.	<GNR> = Device number
TBO	MW addr.	<TBO> = Upper limit text block address
ZA	MW addr.	<ZA> = Number of characters in all
ZE	MW addr.	<ZE> = Number of characters echo
WT	MW addr.	<WT> = Wait time; unit = 10 msec
RDY	Bit addr.	Ready signal; 0 = reading in; 1 else
SBO	MW addr.	<SBO> = Text block address for call; <SBO>-1 = Number
MRY	MW addr.	Internal memory
FN	MW addr.	Error code

TZ Dead Time Element (SFB355)

Formal operand	**Identi-fier**	**Meaning**
TZ		Operation (call)
STAT	Word addr.	Status control loop: -2 = Halt, -1 = Reset, 0 = Initial, 1 = Running
RST	Bit addr.	Reset operating mode (”1” = Reset)
HALT	Bit addr.	Halt operating mode (”1” = Halt)
TT	FWord addr.	Dead time in seconds
X	FWord addr.	Input
YRST	FWord addr.	Reset value output
Y	FWord addr.	Output
ALT	Bit addr.	”1” = no previous values
AF	Bit addr.	AF = 1: error
WAF	Word addr.	Error code
VI	VTZ?	Data structure organization information

UZONE Integrity, Dead Zone (SFB291)

Formal operand	**Identi-fier**	**Meaning**
UZONE		Operation (call)
EF	Bit addr.	Enable
GE	MF addr.	Input quantity
GK	MF addr.	<GK> = Gradient outside of neutral zone
GUZ	MF addr.	<GUZ> = Half-width of neutral zone
GKZ	MF addr.	<GKZ> = Gradient inside of neutral zone
GA	MF addr.	Output quantity
AF	Bit addr.	AF=1: error
WAF	MW addr.	Error code

VAB Switch OFF-Delay 100 msec Pulse (SFB106)

Formal operand	Identi-fier	Meaning
VAB		Operation (call)
E	Bit addr.	Input
WS	MW addr.	<WS> = Setpoint value
WI	MW addr.	<WI> = Actual value
A	Bit addr.	Output

VAL Switch ON-Delay, 1 sec Pulse, Holding Input (SFB107)

Formal operand	Identi-fier	Meaning
VAL		Operation (call)
E	Bit addr.	Input
EH	Bit addr.	Hold input
WS	MW addr.	<WS> = Setpoint value
WI	MW addr.	<WI> = Actual value
A	Bit addr.	Output

VAN Switch ON-Delay, 100 msec Pulse (SFB105)

Formal operand	Identi-fier	Meaning
VAN		Operation (call)
E	Bit addr.	Input
WS	MW addr.	<WS> = Setpoint value
WI	MW addr.	<WI> = Actual value
A	Bit addr.	Output

VBS Compare Two Bit Strings (SFB111)

Formal operand	**Identi- fier**	**Meaning**
VBS		Operation (call)
EF	Bit addr.	Enable
WBI	MW addr.	<WBI> = Start of 1st marker string (actual value)
WN	MW addr.	<WN> = Length of marker string
WBV	MW addr.	<WBV>= Start of 2nd marker string (comparison value)
AA	Bit addr.	Marker strings: equal (AA = 1), unequal (AA = 0)
AF	Bit addr.	AF = 1: error

VIP+ Communications Block for Viewstar 200 XA (SFB1)

Formal operand	Identi-fier		Meaning
VIP+			Operation (call)
EF	Bit addr.		EF = 1: Enable
SP	TN addr.		Sample number in the equipment list (1 ... 10)
VVZ	VVZ?		Data structure ”VIP-Directory”; the current VS200 station name assigned in the AKF SYM/COM editor to the sample of the data structure VVZ is entered here.
VVI	VVI?		Data structure ”internal organization information”
AF	Bit addr.		AF = 1: error
WAF	MW addr.		Error word

VIPS+ Communications Block for Viewstar 200PC (SFB2)

Formal operand	Identi-fier		Meaning
VIPS+			Operation (call)
EF	Bit addr.		EF = 1: enable
KEF	Bit addr.		Trigger for curve data; send if 0 $\rightarrow$ 1 edge
BEF	Bit addr.		Trigger for image data; send if 0 $\rightarrow$ 1 edge
ABF	M addr.		Receiver bit for job/command message
STG	MW addr.	52	<STG> = first word of commission message
MTG	MW addr.	52	<MTG> = first word of signal message;
VVZ	VVZ?		Data structure ”VIP-directory”; the current VS200 station name allocated in the AKF SYM/COM editor to the sample of the data structure VVZ is entered here.
VVI	VVI?		Data structure ”internal organization information”
MSF	M addr.		MSF = 1: Send; Sending bit for communications messages
AF	Bit addr.		AF = 1: error
WAF	MW addr.		Error word

VWS Compare Two Word Strings (SFB110)

Formal operand	Identi-fier	Meaning
VWS		Operation (call)
EF	Bit addr.	Enable
WBI	MW addr.	<WBI> = Start of 1st word string (actual value)
WN	MW addr.	<WN> = Length of the word string
WBV	MW addr.	<WBV>= Start of 2nd word string (comparison values)
AA	Bit addr.	Word strings: equal (=1), unequal (=0)
AF	Bit addr.	AF = 1: error

WAG Convert ASCII to Floating Point Word (SFB219)

Formal operand	Identi-fier	Meaning
WAG		Operation (call)
EF	Bit addr.	Enable
BT	Bit addr.	Commission: convert if 0/1-edge
TBO	MW addr.	<TBO> = Upper limit text block address
AZ	Bit addr.	Previous state of signal BT
WA	MF addr.	Word for converted value
AF	Bit addr.	AF = 1: error

WAH Convert ASCII to Word or Double Word (SFB205)

Formal operand	Identi-fier	Meaning
WAH		Operation (call)
EF	Bit addr.	Enable
BT	Bit addr.	Commission: Convert if 0/1-edge
TBO	MW addr.	<TBO> = Upper limit text block address
K1	Bit addr.	0 = convert word; 1 = convert double word
AZ	Bit addr.	Previous state of signal BT
WA	MW addr.	Word for converted value
AF	Bit addr.	AF = 1: error

WDE Convert Double Word to Word (SFB146)

Formal operand	Identi-fier	Meaning
WED		Operation (call)
WE	MD addr.	Source
WA	MW addr.	Target
AF	Bit addr.	AF = 1: error

WDN Convert BCD -20 Bits- to Word BCN (SFB149)

Formal operand	Identi-fier	Meaning
WDN		Operation (call)
EF	Bit addr.	Enable
BK	Bit addr.	Smallest bit address of bit string
BL	Bit addr.	Last bit address of bit string
EV	Bit addr.	Sign input (1 = negative)
WA	MW addr.	<WA> = Number converted into BCN-code

WED Convert Word to Double Word (SFB145)

Formal operand	Identi-fier	Meaning
WED		Operation (call)
WE	MW addr.	Source
WA	MD addr.	Target

WEIN Direct Input from a Pin String to a Word (SFB189)

Formal operand	Identi-fier	Meaning
WEIN		Operation (call)
EF	Bit addr.	Enable
EK	Bit addr.	Smallest bit address of bit string (I/O)
WA	MW addr.	<WA> = Binary values of the 16 signals

WND Convert Word BCN to BCD -20 Bits- (SFB147)

Formal operand	Identi-fier	Meaning
WND		Operation (call)
EF	Bit addr.	Enable
WE	MW addr.	<WE> = BCN number to be output
BK	Bit addr.	Smallest bit address of bit string
BL	Bit addr.	Last bit address of bit string
AV	Bit addr.	Sign output (1 = negative)

WOSAx Word Collector Words (SFB170-172)

Formal operand			Identi-fier	Meaning
WOSA4	WOSA8	WOS16		Operation (call)
EF	EF	EF	Bit addr.	Enable
WE1	WE1	WE1	MW addr.	1st input M word
WE2	WE2	WE2	MW addr.	2nd input M word
WE3	...	...	...	
WE4	WE8	WE16	MW addr.	xth input M word
WA	WA	WA	MW addr.	Target - M word for WE1

WOVEx Word Distributor Word (SFB176-178)

Formal operand			Identi-fier	Meaning
WOVE4	WOVE8	WOV16		Operation (call)
EF	EF	EF	Bit addr.	Enable
WE	WE	WE	MW addr.	Address 1st M word of source word string
WA1	WA1	WA1	MW addr.	Target - M word for WE
WA2	WA2	WA2	MW addr.	Target - M word for WE+1
...	...	...	...	
WA4	WA8	WA16	MW - addr.	Target - M word for WE+(x-1)

WXOR Word Exclusive-OR (SFB102)

Formal operand	Identi-fier	Meaning
WXOR		Operation (call)
EF	Bit addr.	Enable
WU	MW addr.	<WU> = 1st M word address for logic operation
WN	MW addr.	<WN> = Number of M words for logic operation
WA	MW addr.	<WA> = Result of logic operation

ZA05 Transmit ZAE 105 Data to PLC (SFB35)

Formal operand	Identi-fier	Number	Meaning
ZA05			Block call
EF	Bit addr.	1	Enable
ER	Bit addr.	1	Reset block and load standard values
SP	MW addr.	1	<SP> = Slot reference (slot); 2 ... 160
UF	MD addr.	5	Scaling factor; clock pulse number is divided by the specified value and stored in IW.
VI2	MW addr.	2	Organization information
AAK	Bit addr.	5	AAK = 1: counter activated
IW	MD addr.	5	Counter value
VA1E	Bit addr.	5	Value of prestop 1; negation of outputs (NAUS) is ignored
VA2E	Bit addr.	5	Value of prestop 2; negation of outputs (NAUS) is ignored
EAE	Bit addr.	5	Value of endstop; negation of outputs (NAUS) is ignored
VLSE	Bit addr.	5	VLSE = 1: premature load/start
VEAE	Bit addr.	5	VEAE = 1: pre-endstop
RSE	Bit addr.	5	RSE = 1: clock pulse input enable
EE	Bit addr.	7	Value of process inputs 1 ... 7 of ZAE 105
AP1	Bit addr.	1	AP1 = 1: setpoint value field changed
AP2	Bit addr.	1	AP2 = 1: initialization field changed
AF	Bit addr.	1	AF = 1: error
WAF	MW addr.	1	Error code

ZA05E Simplified Parametrization for ZAE 105 (SFB37)

Formal operand	Identi-fier	Number	Meaning
ZA05E			Operation (call)
EF	Bit addr.	1	Enable
ER	Bit addr.	1	Reset block and load standard values
ES	Bit addr.	1	Write access enable for the RS 232 C interface of the ZAE 105 ES = 1; data can be changed with the RS 232 C interface
EQ	Bit addr.	1	Short circuit acknowledgement for 0 → 1 edge
SP	MW addr.	1	<SP> = Slot reference (slot); 2 ... 160
UF	MD addr.	5	Scaling factor
LS	Bit addr.	5	Load and start counter for 0 → 1 edge
VE	MD addr.	5	Preset value
VA1	MD addr.	5	Prestop value 1; presetting only possible for counters 1 ... 3
IEA	Bit addr.	5	IEA = 1; endstop input enable
VI3	MW addr.	3	Organization information
RDY	Bit addr.	1	Ready signal; RDY = 0 during data transmission between PLC and ZAE 105
AF	Bit addr.	1	Error bit; AF = 1 if error occurred
WAF	Word addr.	1	Error code; <WAF> = error number

ZA05I Parametrization for ZAE 105 (SFB36)

Formal operand	Identi-fier	Number	Meaning
ZA05I			Operation (call)
EF	Bit addr.	1	Enable
ER	Bit addr.	1	Reset block and load standard values
UEBR	Bit addr.	1	Transmission direction: UEBR = 0; ZAE 105 → controller, UEBR = 1; controller → ZAE 105
ES	Bit addr.	1	Write access enable for the RS 232 C interface of the ZAE 105 ES = 1; data can be changed with the RS 232 C interface, transmission direction only controller → ZAE 105
EQ	Bit addr.	1	Short circuit acknowledgement for 0 → 1 edge Transmission direction only controller → ZAE 105
SP	MW addr.	1	<SP> = Slot reference (slot); 2 ... 160
UF	MD addr.	5	Scaling factor

Continued on next page

Formal operand	Identi-fier	Number	Meaning
ZA05I continued			
EFB1	Bit addr.	1	Block enable 1 (setpoint value field)
STRT	Bit addr.	5	Start counter for 0 → 1 edge, taking UVO and IS into consideration
LS	Bit addr.	5	Load and start counter for 0 → 1 edge, taking UVO and IS into consideration
RS	Bit addr.	5	Clock pulse input enable (software gate function) Clock pulses are only counted if RS = 1 and the hardware gate (TAE) is open
VE	MD addr.	5	Preset value
VA1	MD addr.	5	Prestop value 1
VA2	MD addr.	5	Prestop value 2
EA	MD addr.	5	Endstop value
IMP	MW addr.	5	Clock pulse monitoring time for sensor pulses <IMP> = 1 ... 255 (x 100 msec) <IMP> = 0; no clock pulse monitoring
BEA	Bit addr.	5	Pre-endstop; BEA = 1: Activate prestop and endstop
BRES	Bit addr.	5	Reset counter: BRES = 1
EFB2	Bit addr.	1	Block enable 2 (initialization data)
AK	Bit addr.	5	Activate counter; only the activated counters of the ZAE 105 are processed, AK = 1; counter activated
PK	Bit addr.	5	Parametrize counter; the initialization data are only accepted for the counter for which PK = 1.
EBUA	Bit addr.	5	Switch-off behavior of the ZAE 105 after expiration of the monitoring time EBUA = 1; activate prestop and endstop EBUA = 0; retain function of the ZAE 105
TUE	MW addr.	1	Monitoring time 1 ... 255 x 100msec; the reaction defined in EBUA occurs after expiration of the monitoring time
BETR	MW addr.	5	Operating mode; 1 = result counter 1 with parallel prestop 2 = result counter 2 with serial prestop 3 = differential counter 1 with parallel prestop; only permitted for counter 1/3 4 = differential counter 2 with serial prestop; only permitted for counter 1/3 5 = repetition counter 6 = speed counter with T = 1 sec 7 = speed counter with T = 10 sec

Continued on next page

Formal operand	Identi-fier	Number	Meaning
ZA05I continued			
EAS	Bit addr.	5	Endstop mode; EAS = 0; normal Endstop; 5 msec typical, EAS = 1; fast Endstop; 200 µsec typical
VAR	Bit addr.	5	Prestop mode: VAR = 0: absolute values; VAR = 1: relative values
TAE	Bit addr.	5	Assign gate; HW gate function enable or inhibit: TAE = 0: without HW gate; TAE = 1: with HW gate
NC	Bit addr.	5	Negate counter input (clock); NC = 1; negate counter input
NE	Bit addr.	7	Negate process input; NE = 1: negate input
UVO	Bit addr.	5	Logic operation for the start inputs defined with IS: UVO = 0: AND operation for inputs UVO = 1: OR operation for inputs
IS	MW addr.	5	Input selection for the load/start (LS) or start (STRT) commands of the counter. Maximum 3 inputs per counter
IER	MW addr.	5	Input selection for BRES (reset counter) and BEA (pre-endstop), 1 input for BRES per counter, maximum 3 inputs for BEA per counter.
AUSA	MW addr.	5	Output selection for VA1, VA2 and EA
NAUS	MW addr.	5	Selection of outputs to be negated
VI3	MW addr.	3	Organization information
RDY	Bit addr.	1	Ready signal; RDY = 0 during data transmission between PLC and ZAE 105
AF	Bit addr.	1	Error bit; AF = 1 if error occurred
WAF	MW addr.	1	Error code; <WAF> = error number
AW	Bit addr.	1	Warning bit; AW = 1 if a warning occurred
WARN	MW addr.	1	Warning code; <WARN> = number of warning

Formal operand	Identi-fier	Number	Meaning
ZR			Operation (call)
STAT	Word addr.		Status control loop: -2 = Halt, -1 = Reset, 40 = Initial, 1 = Running
BT	BZR?		Data structure operating modes (see below)
PARA	PZR?		Data structure parameter (see below)
W	FWord addr.		Setpoint value input
X	FWord addr.		Actual value input
NG	FWord addr.		Standardization quantity
XRR	FWord addr.		Reset value of feedback (%)
YHND	Bit addr.		Manual value for Y
Y	Bit addr.		Output manipulated variable
AF	Bit addr.		AF = 1: error
WAF	Word addr.		Error code
VI	VIC?	48 Byte	Data structure organization information

BZR

Element	Element type	Symbol suggestion	Meaning
BZRn			Data structure operating modes of ZR, n = 100
BZRn.1	Bit	Reset	Control input for reset operating mode (”1” = Reset)
BZRn.2	Bit	Hand	Control input for manual operating mode (”1” = Hand)
BZRn.3	Bit	Halt	Control input for halt operating mode (”1” = Halt)
BZRn.4	Bit	PID-Par	Use of PID-parameter (PID-Par = 1)

PZR

Element	Element type	Symbol suggestion	Meaning
PZRn			Data structure parameters of ZR, n = 50
PZRn.1	Float word	Kp	Proportional rate (gain)
PZRn.2	Float word	Tn	Reset time
PZRn.3	Float word	Tv	Derivative action time
PZRn.4	Float word	Kr	Feedback gain
PZRn.5	Float word	T1	Time constant of high-speed feedback
PZRn.6	Float word	T2	Time constant of low-speed feedback
PZRn.7	Float word	HYS	Hysteresis of two-position switch

ZVR Forwards-Backwards Counter Word (SFB103)

Formal operand	Identi-fier	Meaning
ZVR		Operation (call)
E	Bit addr.	Dynamic counter input (0/1-edge)
ER	Bit addr.	Reset actual value: 0 $\rightarrow$ <WI>
EF	Bit addr.	Enable
EZ	Bit addr.	Count direction
WS	MW addr.	<WS> = Setpoint value
AZ	Bit addr.	Previous signal state
WI	MW addr.	<WI> = Actual value
AV	Bit addr.	Output sign of WI
WD	MW addr.	<WD> = differential value
AD	Bit addr.	Output sign of WD
AG	Bit addr.	Output “Greater”, (<WI> > <WS>)
AA	Bit addr.	Output “Equal”, (<WS> = <WI>)
AF	Bit addr.	AF = 1: Error

Part V
Dolog AKF for Beginners

Chapter 1
Introduction

This chapter gives you a short survey of the components for programming with Dolog AKF. The basic functions of the software are also defined.

1.1 General Information

The Dolog AKF software is used for structured programming of PLC user programs using modern window techniques.

Programs are created and displayed in three special languages, the instruction list, the ladder diagram and the function block diagram (see also Standardization Draft IEC 65A(SEC)65).

The programs contain different types of blocks which are combined depending on their use and on the complexity of the task. The organization blocks, programming blocks, function blocks, standard function blocks and SYM/COM blocks here have different tasks (see also chapter 4 in this part):

- Organization of the complete program
- Combination of technological program parts
- Simplification of program repetition (”subroutines”)
- Simplification of program with predefined program parts
- Inclusion of symbolic programming

After a short introduction to ”structured programming” with its program parts, AKF35 will be described in depth in further chapters. An user example in which the ”first steps” can be practiced follows the short summary of the features. A job description for the example to be found on your AKF35 original diskette is also provided.

1.2 Programming Components

What do you need to program your programmable controller?

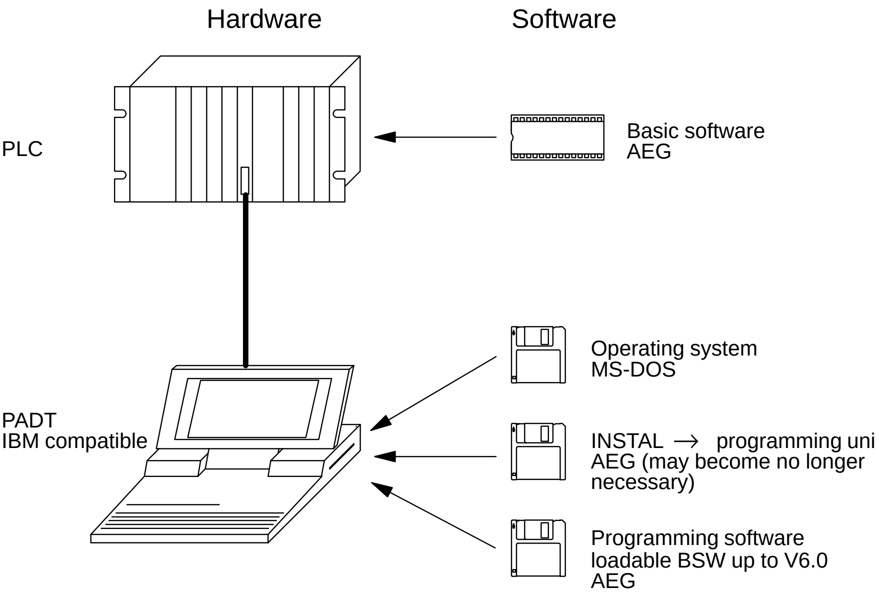


Figure 21 Components for Programming a Programmable Controller

1.3 Basic Functions

User programs are created off-line in Dolog AKF and then transferred to the programmable controller. On-line functions, e.g. a view of the program during the scan, are then possible.

The following basic functions are carried out with Dolog AKF:

❐ **Edit** (Create / Modify)	-	off-line
❐ **Load** (to / from PLC)	-	off-line or on-line
Compare (PADT / PLC)	-	on-line
❐ **Online**	-	on-line
❐ **Print**	-	off-line
❐ **Special**	-	off-line
❐ **Setup**	-	off-line

Chapter 2
Structured Programming

After a general explanation, the different block types in Dolog AKF will be explained. The structure levels are illustrated with a figure.

2.1 General Information

The performance and economic efficiency of a programming system depends on a number of conceptual features.

Structural and standardization features, the use of a universal personal computer as programming panel (PADT) and an easy-to-use operator interface add up to advantages in keeping the software and maintenance costs as low as possible.

Keeping in mind the amount of information and program volumes common today in programmable controllers, program sections also help optimize scan times. Time-critical processes require fast reactions. These can be obtained with skillful configuration by not executing program sections which need not be processed.

Subdividing a job complex makes the overall problem comprehensible.

Program sections are easier to create and to test if they are individual self-contained software blocks. On the other hand, the clarity suffers if related flows are integrated in copious ”monolithic” overall programs.

Configuration with easy-to-use programming software saves the user unclear jumps.

2.2 Program Structure

The special languages permit the structuring and input of programs. Program input and display are possible in the instruction list, ladder diagram and function block diagram.

Structuring of a program means the creation of transparent, comprehensible, self-contained user program sections, the so-called blocks.

Technology-dependent, repetitive flows can be used repeatedly as tested program sections within a system or as technological blocks. Function blocks can be combined to form both universal and user-dependent program libraries. Standard function blocks integrated in the PLC for complex control, data handling and operating functions form the basis of a simple structure of complex user-dependent software blocks.

The blocks are combined from networks. These represent the lowest structure level. The logic is contained in the networks. It contains operations to which parameters are assigned (in the instruction list also called instructions).

The following five block types are used for assignment in a job complex (see chapter 4 in this part).

The following figure shows an example of the different structure levels.

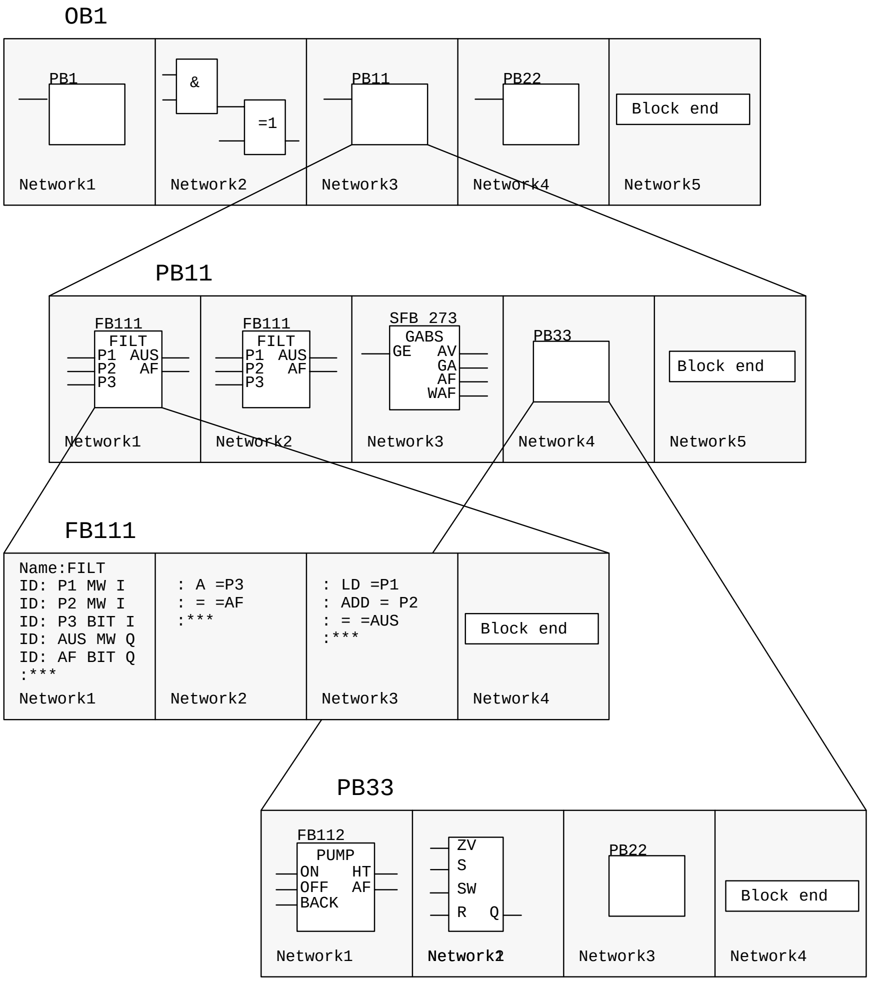


Figure 22 Example of a Section of a Structured Program

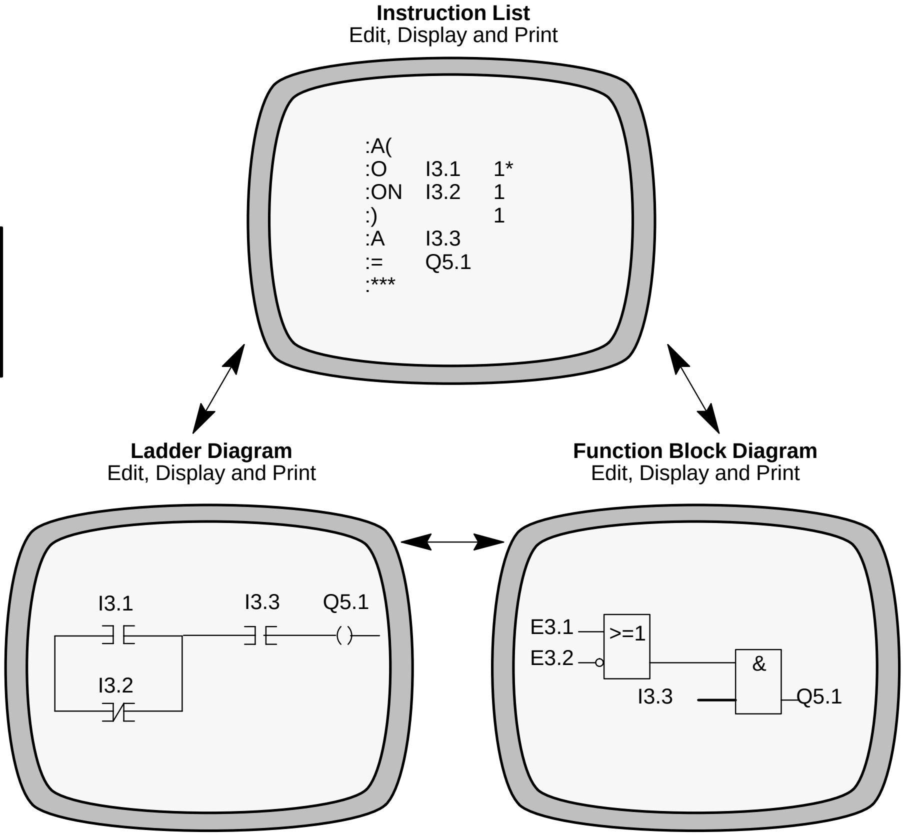
Chapter 3
Special Languages of Programming

This chapter gives a short explanation of the individual special languages
Instruction list IL
Ladder diagram LD and
Function block diagram FBD.

The Dolog AKF software is based on the system of structured programming in the standardized special languages. Please read DIN 19239 or Standardization Draft IEC 65A(SEC)65 about the standardized definitions (structure etc.).

- Instruction list (IL)
- Ladder diagram (LD)
- Function block diagram (FBD)

It is possible to display the blocks programmed in one of the individual special languages in another special language.



*1 is the number of brackets used

Figure 23 Display in Different Special Languages

3.1 Instruction List IL

The instruction list is a standardized display mode in alphanumeric form.

The instructions are strung together line-by-line when creating programs in IL.

There are two different kinds of instructions. Both of the following instructions can be used in the same manner (AF can be selected as required):

: O I3.1   : O =AF

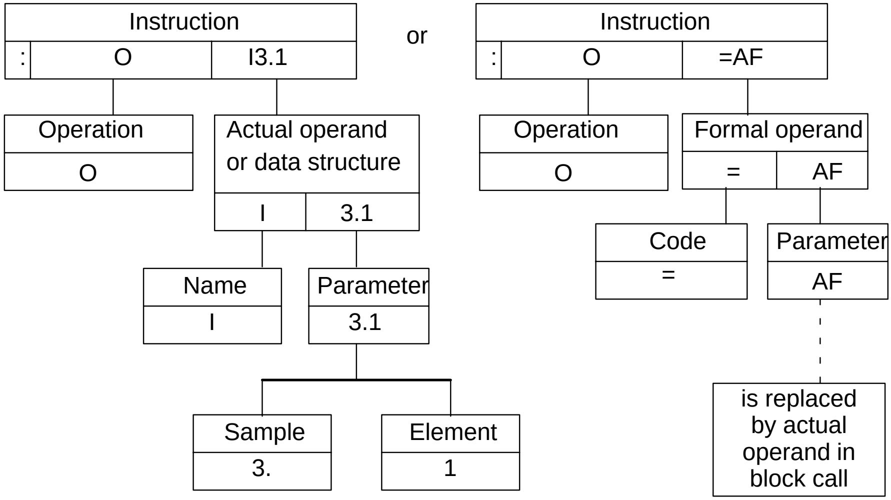


Figure 24 Explanation of Terminology in Instruction List

The instruction list of a network is terminated by the end-of-network character "***".
The end of a block is identified by "BE", Block End.

Tables of the operations and operands/data structures exist for the individual controllers (for information about data structures see also chapter 4.6).

Organization blocks (OBs), program blocks (PBs) and function blocks (FBs) can be programmed in the IL.

Jumps and block calls are possible in the instruction list.

☞ **Note** Detailed information for example about creating programs in the instruction list can be found in the section ”Configuration”.

3.2 Ladder Diagram LD

The ladder diagram is a standardized graphic display mode.

The following basic symbols can be used when creating ladder diagrams.

—] [—	Normally open contact
—]/[—	Normally closed contact
———┤	Connection of parallel ladder diagram lines
———	Continuation in parallel path without contacts
—( )—	Output

The ladder diagram operations are parametrized with operands/data structures (for information about data structures see also chapter 4.6).

For an explanation of the terminology see Figure 24.

The end of a block is identified by a square with ”Block End”.

OBs and PBs can be programmed in the ladder diagram.

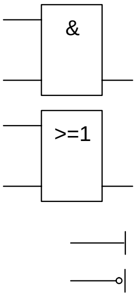
Jumps are not possible, but block calls are possible in ladder diagram.

☞ **Note** Detailed information for example about creating programs in ladder diagram can be found in the section ”Configuration”.

3.3 Function Block Diagram FBD

The function block diagram is a standardized graphic display mode.

The following basic symbols can be used when creating function block diagrams.



AND block

OR block

Input
Input negated

The function block diagram operations are still parametrized with operands/data structures (for information about data structures see also chapter 4.6).

OBs and PBs can be programmed in the FBD.

The end of a block is identified by a square with ”Block End”.

For an explanation of the terminology see Figure 24.

No jumps are possible, but block calls are possible in the function block diagram.

☞ **Note** Detailed information for example about creating programs in function block diagram can be found in the section ”Configuration”.

Chapter 4
Blocks

The following chapter describes the most important functions and features of the various types of blocks.

4.1 Block Types

- The **organization block OB**
  contains the rough structure and defines the order in which the further blocks are to be processed.

- The **program block PB**
  combines user program parts according to technological aspects such as modules, machine parts and plant sections.

- The **function block FB**
  processes program parts which occur frequently as separate subroutines.

- The **standard function block SFB**
  has the same task as the FB and is an integrated element of the standard PLC functions.

Individual networks in the display as IL, LD or FBD are the ”substructures” contained in program blocks and function blocks (FBs and SFBs). The individual networks make up the program with the instruction sequence for the particular process control.

The user program in Dolog AKF contains different blocks. These blocks are selected depending on the complexity of the problem and to minimize the configuration costs. The block technology therefore is adapted to structured programming.

- The **SYM/COM block**
  contains and organizes the allocation of the hardware addresses, symbolic addresses and comments. It cannot be linked into a network and is created separately with the SYM/COM editor.

4.2 Organization Block OB

Two types of organization block are provided in the AKF35.

- OB1
  The OB1 defines the structure for the complete user program.

- OBi of OB2 to OB999
  If closed-loop control is used, the control loops are in the OBi (OB2 to OB999). The OB1 defines when the closed-loop control OB is called.

Organization blocks can be created in IL, LD or FBD.

The OB is processed scan-by-scan. Each scan begins with the processing of network 001 and ends with the processing of the last network contained in the OB.

4.2.1 OB1

The organization block OB defines the structure for the complete user program.

The program and function blocks PB, FB and SFB are called and processed by the OB in the required order.

PBs and FBs are strung together here in a sequence of networks which are numbered continuously beginning with network 001.

Each network contains only one PB, FB or SFB call (except for IL) or one user program segment in IL, LD or FBD.

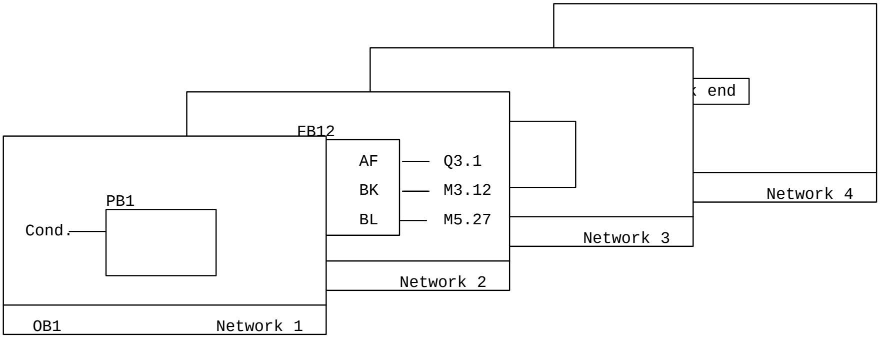


Figure 25 Example of Networks in an Organization Block

The processing of a block (PB, FB or SFB) is followed by the corresponding block call, which may depend on a condition. This is followed by a jump back to the next network of the OB.

4.2.2 OBi

A block for managing the closed-loop control is selected from OB2 to OB999.

The user program is entered in a standard function block, SFB 390. SFB 390 is called once in the OB1. The required control organization block is entered in the formal operand OB (e.g. OBi = OB4).

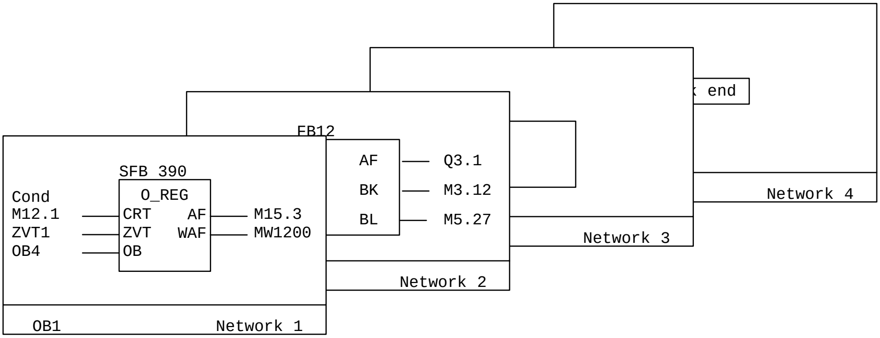


Figure 26 Example of Control OB (OBi, here OB4) in the OB1

4.3 Program Block PB

A program block generally contains user program segments which belong together technologically, e.g. one of x different machines.

Program blocks can be created in IL, LD or FBD.

Structure:
A PB comprises a string of networks numbered continuously beginning with network 001. You can create IL, LD or FBD program parts or call conditional or unconditional PBs, FBs and SFBs in the networks.

Call:
PBs are called by the OB, another PB or a FB.

You can call one and the same PB more than once.
A block to be called is displayed in the network as a square (in LD/FBD). The PB number is above the square. In conditional PBs, the signal address of the condition is to the left next to the square.
A PB which is not called anywhere is never processed.

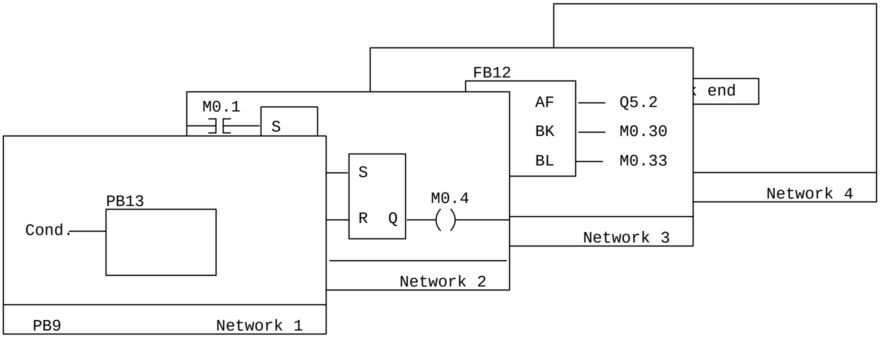


Figure 27 Example of Networks in a Program Block

4.4 Function Block FB

FBs are used to create program segments which are repeated frequently. They are parametrizable subroutines, i.e. a FB can be called and parametrized more than once at different locations.

Function blocks can only be created in IL.

You must distinguish between the function block and the call of a function block. The function block contains an user program segment. The call of a FB ensures that the FB is processed during runtime exactly when it is activated by the corresponding call in the user program. The parametrization of the FB is passed to the subroutine (formal operands are replaced by actual operands/data structures) before the block is processed. An FB which is not called anywhere is never processed.

Structure:
The program of the function block contains a declaration and an instruction part.

- **Declaration part**
  The declaration part is always in the first network of an FB.
  Enter the name of the function block and the list of the formal operands, specifying the type, in the declaration part.
  The declaration part also contains information about the graphic structure of the block square and the order of the parameters.
  A limited number of alterations of the declaration part is permitted at a later time.

- **Instruction part**
  You create the instruction list, the logical and algorithmic relationships between the formal operands, in the declaration part of the instruction part.
  The names of the formal operands must always be preceded by a ”=”-character in the instruction list.
  A number specifying the nesting depth of the relevant line may be included to the right of the instruction list.

Call

```
                FB1
             +---------------+
Condition    |    EXAMPLE    |
BIT?         | EN        RES |  MW?
WORD?        | OP1       ERR |  BIT
WORD?        | OP2           |  ?
             +---------------+
```

Conditional call of a function block in ladder diagram or function block diagram. You see the formal operands in the block.
Actual operands/data structures (M3.3, Q3.1, ZVT1...) should be inserted in place of the question marks.

Figure 28 Example of a Conditional Call

A FB can be called from another FB, a PB or an OB. The FB is then displayed in LD/FBD as a square in the network.

Its name is displayed again in abbreviated form in the square. The input formal operands are to the left inside the square and the actual operands/data structures are to the left outside the square. A condition may be above the square, the output formal operands are at the right in the square and the actual operands/data structures are to the right outside the square. You need only enter the parameters outside the square after calling the FB.

If you alter the formal operands in the declaration part of a FB, you must parametrize all the affected FB calls again. You can easily determine the locations of the FB calls of the particular FBs in the user program with the program overview or the global cross-reference list.

You can also call a FB in the instruction part of another function block (nesting, recursion).The function block called in this way may contain the formal operands of the calling FB as actual operands.

4.5 Standard Function Block SFB

A library of standard function blocks is provided with your Dolog AKF software. These blocks are already defined and need only be called (conditionally or unconditionally) and parametrized by the user at the required location.

The declaration part and the instruction part of the SFB already exist in the software and may not be altered by the user. The formal operands are predefined. If you are configuring, call the block at the required location in the program and parametrize it with actual operands or data structures of your choice (see also chapter 4.6).

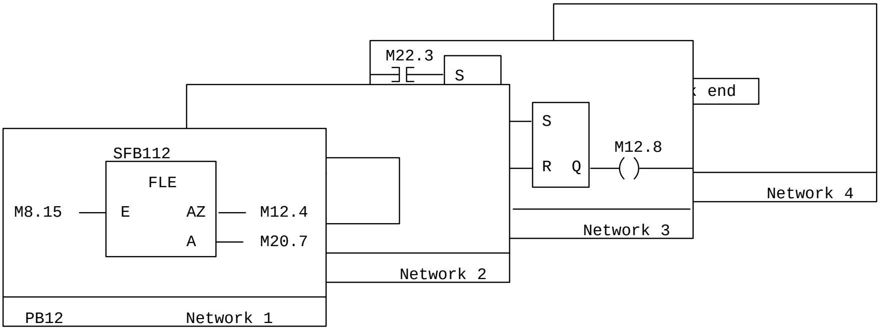


Figure 29 Example of an SFB Call in the Network of a PB (NW1)

4.6 Data Structures

Data structures are provided in AKF35 as of version 6.0.

The data structure is a table of parameters which belong together technologically. One distinguishes between:

- A: Operands (actual operands) which are already familiar from earlier AKF versions

- B: Data structures defined by AEG and used to parametrize SFBs and intelligent function modules

- C: Data structures which you create yourself

All data structures can be displayed in the AKF35 menu ”Edit”, ”Data Structures”.

Definitions for the AEG data structure elements are made in the menu ”Edit”, ”Symbols and Comments”.

User data structures are set up and defined in the menu ”Edit”, ”Data Structures”. The values of the elements are defined in the menu ”Edit”, ”Symbols and Comments”.

The following table and example are used to distinguish the different groups.

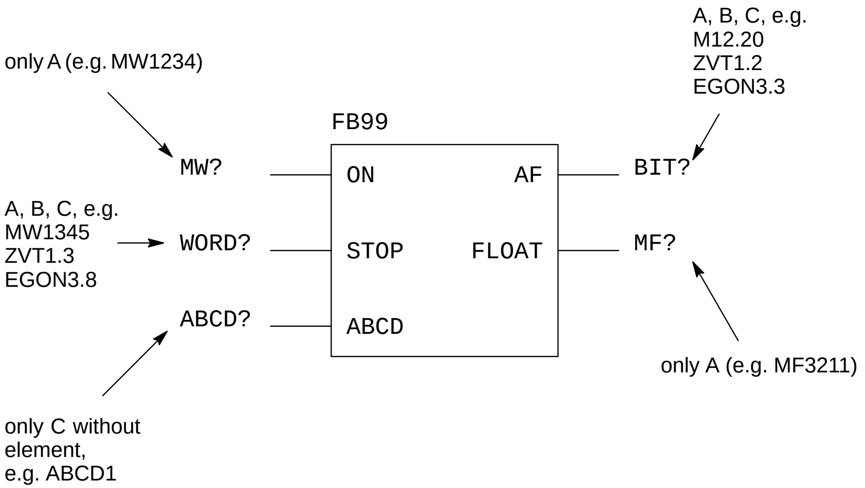
Table 16 Definition of Characteristics of Data Structures

Criterium	A	B	C
Programmer	AEG	AEG	User
Category	general	user-dependent	user-dependent
in the signal memory	yes	no	no
in the AKF35 data structure Editor	look at	look at	create
Names	1...2 letters	3...4 letters	at least 4 letters
e.g.	I, Q, MW	ZVT, APOA	HUGO, EXAMP
Element types*	BIT, WORD, DOUBLE WORD, FLOAT WORD	BIT, BYTE, WORD, DOUBLE WORD, FLOAT WORD, POINTER,STREAM	BIT, BYTE, WORD, DOUBLE WORD, FLOAT WORD, POINTER, STREAM
Use	only alone	mixed possible	mixed possible
Structure	I, Q, M: NameSample.Element MW, MD, MF: Name1...10000	NameSample.Element	NameSample.Element
e.g.	I3.8, MW999	ZVT1.3, ZVT1	EGON2.7, EGON2
can be used in FB as formal operand	all	only single elements	only single elements
e.g.	A M12.20 corresp. to A =OPA	A ZVT1.3 corresp. to A =OMA	A EGON2.7 corresp. to A =EVA

* in data structure editor; display of type A in FB or SFB: I, Q, M, MW, MD, MF

☞ **Note** A table of groups A and B can be found in the ”Programming” section in the chapter ”Validity Scopes and System Operands”.

The following FB is an example containing all types of data structures. The allocation of formal operands and data strutures can be found in the declaration part of the function block. In this case only the FB call is specified.



☞ **Note** Standard function blocks which already existed before version 6.0 may only be parametrized with type A (operand).

☞ **Note** A table of the AEG data structures in the SYM/COM block can be found in part III, ”Configuration” in the chapter ”Validity Scopes and System Operands”.

4.7 SYM/COM Block

You can assign symbolic names and comments to absolute addresses to make the relationship between an absolute address (input/output, marker etc.) and its technological function clear.
The text of the symbolic names and comments is stored in the SYM/COM block under the current station name.
After activating the SYM/COM block, the symbolic names entered in the SYM/COM block can be used in place of the absolute addresses during programming.
The SYM/COM block can also be documented.

Signal	Symbol	Initial value	Comments >>
I3.1	ON		Motor 1 on
I3.2	MOT_RI		Motor right on
I3.3	MOT_LE		Motor left on
I3.4	STOP		Emerg. stop button
I3.5	PUMP_1		Pump 1 on
I3.6	PUMP_2		Pump 2 on
I3.7			
I3.8			
I3.9			
I3.10			
I3.11	GRAB_UP		Grabs up
I3.12	GRAB_DWN		Grabs down
I3.13			
I3.14			
I3.15			
I3.16			

Figure 30 Example of the Entries in the SYM/COM Block

Chapter 5
First Programming Steps with AKF35

This chapter contains a small example of an application of AKF35 describing all aspects. It was processed with software version 6.0.

5.1 General Information

The first programming steps for AKF35 beginners are described in this chapter. A simple program is programmed in AKF35, transferred to the PLC and then looked at with the dynamic status display.

5.2 Preparatory steps

You should already have carried out the following preparatory steps:

- You installed the software on drive C: of your programming panel (PADT, here P610) (see part II)

☞ **Note** There is no Modnet 2/NP or Modnet 1/SFB installation for the example. The example program of the software is also installed.

- An A500 with the following relevant modules is provided for the example: ALU 021 (BSW 6.0) with arithmetic coprocessor, BIK 151, DEA 106, DAP 102 (at slot reference 2) and as simulator two SIM 011s at DAP-inputs I2.17 to I2.32.
  Please do not forget the relevant cable.

5.3 Problem Statement

Create a program for which an 8-bit bit pattern runs on 16 bits of an output module ("running light"). The bit pattern to be used is set with inputs I2.25 to I2.32 and accepted with input I2.18. You can stop the output with I2.17 = 1 (all 16 outputs = 0) or "freeze" the current state with I2.19 = 0. Program in the special language instruction list in DIN addressing and symbolic.

The plant in the example is called "AKF35", the program is called "EXERCISE".

☞ **Note** The logic of the program exists, here it is a matter of practice in using the AKF.

5.4 Parameters of the Programming Example

Table 17 Operands in the Programming Example

Signal	Symbol	Initial value	Comments
I2.17	OFF	–	Off=1: all outp. ”0”, Off=0: display
I2.18	LOAD	–	Bit string is loaded with 0->1 edge
I2.19	EN	–	EN=0: freeze, EN=1: run
I2.25	BIT1	–	First bit of bit string
I2.26	BIT2	–	Second bit of bit string
I2.27	BIT3	–	Third bit of bit string
I2.28	BIT4	–	Fourth bit of bit string
I2.29	BIT5	–	Fifth bit of bit string
I2.30	BIT6	–	Sixth bit of bit string
I2.31	BIT7	–	Seventh bit of bit string
I2.32	BIT8	–	Eighth bit of bit string
Q2.1	RUN1	-	
Q2.2	RUN2	–	
Q2.3	RUN3	–	
Q2.4	RUN4	–	
Q2.5	RUN5	–	
Q2.6	RUN6	–	
Q2.7	RUN7	–	
Q2.8	RUN8	–	Outputs to which the bit pattern
Q2.9	RUN9	–	is output alternately
Q2.10	RUN10	–	(running light)
Q2.11	RUN11	–	
Q2.12	RUN12	–	
Q2.13	RUN13	–	
Q2.14	RUN14	–	
Q2.15	RUN15	–	
Q2.16	RUN16	–	
M1.6	PULSE_5	–	5.0 Hz blinking rate
M1.10	FIXED”0”	–	Constant 0
M1.11	FIXED”1”	–	Constant 1
M15.1	AUX1	0	Aux. marker 1
M15.2	AUX2	0	Aux. marker 2
M15.3	AUX3	0	Aux. marker 3
M15.4	AUX4	0	Aux. marker 4
M15.5	AUX5	0	Aux. marker 5
M15.6	AUX6	0	Aux. marker 6
MW1100	ROTATED	0	This word contains the rotated info
MW1200	SHIFT	1	Shift pulse SFB123

5.5 Programming

Note Menu functions are specified in ”inverted commas”, e.g. ”Edit”, ”Block”. Your (typed) input is written in `Courier`, e.g. `AKF35`. Key combinations/special keys are specified in brackets, e.g. <Ctrl>+<S>. ”Toggle” means press <Return> several times.

5.5.1 Call Program

Step 1 Call the software from user drive C: with `AKF35`

React. The main menu is visible on the screen. The selection bars point to the function which was last used in the last AKF configuration. The bar points to ”Edit” after the installation.

Note You can move the bar to the required menu line with <←>, <→>, <↓>, <↑>. You can call the menu marked by the bar with <Return>.
The marked capital letters (reference characters) have a different color and are used to call the menu directly.
All the steps described below should be executed chronologically (even if numbering begins with ”1” again for each substep).

5.5.2 Set Plant/Station

Step 1 Enter T for ”SeTup”

React. The setup menu is opened

Step 2 Enter L for ”Plant”

React. The plant path of the last configuration is displayed. After installation, C:\AKF35 is predefined.

Step 3 Enter C:\AKF35 and confirm with <Return>

React.
a) If the AKF35 plant does not yet exist, the station name is requested. Enter EXERCISE
b) If the plant already exists, the setup menu is displayed again

Step 4 Enter S for ”Station”

React. The actual setup menu is opened, the selection bar points to ”PC* Station Name”

Step 5 Enable input for ”PC* Station Name” with <Return>

React. The bar is opened in input width

Step 6 Enter EXERCISE

React. A message appears which asks whether you really want to create the plant

Step 7 Enter Y for yes. Acknowledge the message ”No equipment list exists...” with <Return>.

React. The station is now entered

5.5.3 Program Presettings

The following settings are made in the menu ”SeTup”, ”Station” like the station name.

Step 1 Enter `L` for ”ALU Type”

React. A window with the predefined ALU types is opened

Step 2 Move the menu bar to ”ALU021” with the cursor keys and confirm with <Return>.

Step 3 Enter `M` for ”Address Mode”; toggle until ”DIN” appears

Step 4 Enter `A` for ”Addressing”; toggle until ”SYM” appears (symbolic programming)

Step 5 Enter `I` for ”Input Mode”; toggle until ”IL” appears (programming in instruction list)

Step 6 Press <Esc> twice

React. The menues are closed and the bar only points to the main menu line for ”SeTup”. The setup settings are thus accepted.

5.5.4 Edit Equipment List

5.5.4.1 Activate equipment list editor

Step 1 Enter E for ”Edit”

React. The Edit menu is opened

Step 2 Enter Q for ”EQuipment List”

React. The equipment list editor appears on the screen. The first line is standardly empty for ALU type ALU021 (SeTup).

5.5.4.2 Set Subrack

Step 1 Jump to SP2 with <↓> (bar in 2nd column)

Step 2 Open the processing menu with <Ctrl>+<Return> (only possible in 2nd column)

Step 3 Enter F for ”Front Connection”

Step 4 Select subrack ”DTA 102” with <↓> and confirm with <Return> (generally: DTA 102 is always entered for subracks with 4 slots, even if it is a DTA 150)

React. The subrack is entered in the columns (in 4 lines since four slots are available in the subrack)

5.5.4.3 Enter Module

Step 1 Change to ”SP” no. 2 with <→> in the 3rd column

Step 2 Open the module menu with <Return> (only possible in the 3rd column)

Step 3 Select the module ”DAP 102” with <↓> and confirm with <Return>

React. The module now appears in line SP2 in the 3rd column. The comments of the module are specified in the lowest line of the screen. We will now change them.

The following figure shows what your equipment list should now look like.

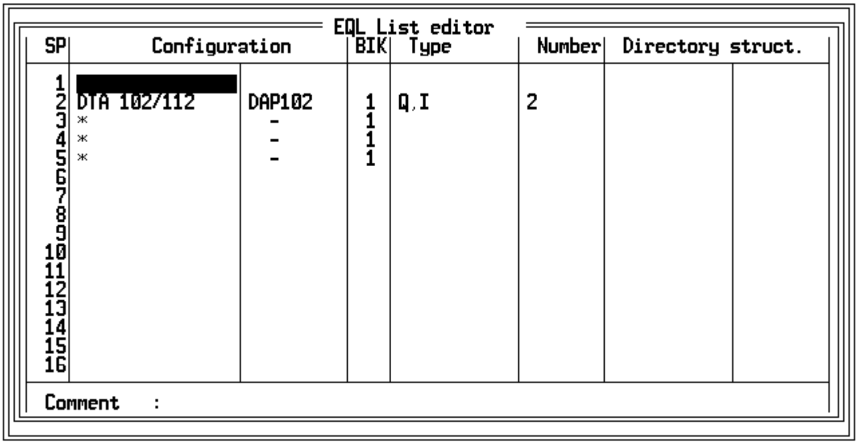


Figure 31 Programming Example of Equipment List

5.5.4.4 Enter/Alter Comments for Module

☞ **Note** You can also carry out a function outside the menu with <Ctrl>+<reference character>. The following steps show an example.

Step 1 Enter <Ctrl>+<S> (for commentS)

React. Cursor points to the lowest line. You can now change the comment.

Step 2 Delete the comment with <Del>

Step 3 Enter the following comment: `This is an input/output module` <Return>

React. The input is terminated with <Return> and the cursor now points to the equipment list editor again

5.5.4.5 Terminate and Store Equipment List

Step 1 You store the equipment list and leave the equipment list editor with <Ctrl>+<T> (for Terminate).

React. You now are looking at the Edit menu again

5.5.5 Assign Symbols and Comments (SYMCOM block)

Since symbolic programming is planned, the signal symbols must be defined. This is best done before creating the program.

Step 1 Enter S for ”Symbols and Comments”

React. The symbols and comments editor appears on the screen

Step 2 Open the processing menu with <Ctrl>+<Return>

Step 3 Enter F for ”Search Function”

React. A window appears in which you can enter the signal to be searched for

Step 4 Enter `I2.17` and <Return>

React. The cursor jumps to the ”Symbol” column of the specified signal in the editor

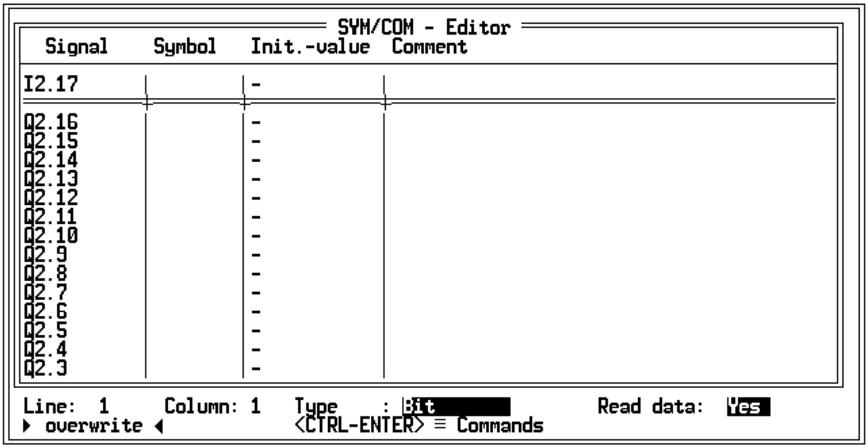


Figure 32 Symbols and Comments Editor

Step 5 Enter `OFF` and <Return>

React. The cursor jumps to the ”Comments” column

Step 6 Enter as comment: `Off=1: all outp. Off=1: display` and confirm with <Return>

React. The cursor jumps to the next line, here Q2.16

Step 7 Move the cursor to I2.18 with <↑>

Step 8 Enter `LOAD` and <Return>

React. The cursor jumps to the ”Comments” column

Step 9 Enter as comment: `Bit string is loaded with 0–>1 edge` and confirm with <Return>

React. The cursor jumps to the next line

Step 10 Move the cursor to I2.19 with <↑>

Step 11 Enter `EN` and <Return>

React. The cursor jumps to the ”Comments” column

Step 12 Enter as comment: `EN=0:freeze, EN=1: run` and confirm with <Return>

React. The cursor jumps to the next line

Step 13 Enter <Ctrl>+<F> for the next search

React. The search window is opened

Step 14 Enter `I2.32` and <Return>

React. The cursor jumps to the ”Symbol” column of the specified signal in the editor

Step 15 Enter `BIT8` and <Return>

React. The cursor jumps to the ”Comments” column

Step 16 Enter as comment: `Eighth bit of bit string` and confirm with <Return>

React. The cursor jumps to the next line (I2.31)

Step 17 Repeat Step 15 and Step 16 with the following input:

`BIT7`	<Return>	`Seventh bit of bit string`	<Return>
`BIT6`	<Return>	`Sixth bit of bit string`	<Return>
`BIT5`	<Return>	`Fifth bit of bit string`	<Return>
`BIT4`	<Return>	`Fourth bit of bit string`	<Return>
`BIT3`	<Return>	`Third bit of bit string`	<Return>
`BIT2`	<Return>	`Second bit of bit string`	<Return>
`BIT1`	<Return>	`First bit of bit string`	<Return>

React. The cursor jumps to the next line (I2.24)

Step 18 Enter <Ctrl>+<F> for the next search

React. The search window is opened

Step 19 Enter `Q2.16` and <Return>

React. The cursor jumps to the ”Symbol” column of the specified signal in the editor

Step 20 Enter `RUN16` and twice <Return>

React. The cursor jumps to the next line (Q2.15)

Step 21 Enter `RUN15` and twice <Return>

React. The cursor jumps to the next line (Q2.14)

Step 22 Repeat Step 21 with the following input
`RUN14` twice <Return>
`RUN13` twice <Return>
`RUN12` twice <Return>
`RUN11` twice <Return>

React. The cursor jumps to the next line (Q2.10)

Step 23 Enter `RUN10` and <Return>

React. The cursor jumps to the ”Comments” column

Step 24 Enter as comment: `(running light)` and confirm with <Return>

Step 25 Repeat Step 23 and Step 24 with the following input

`RUN9`	<Return>	`is output alternately`	<Return>
`RUN8`	<Return>	`Outputs to which the bit pattern`	<Return>

React. The cursor jumps to the next line (Q2.7)

Step 26 Repeat Step 21 with the following input
`RUN7` twice <Return>
`RUN6` twice <Return>
`RUN5` twice <Return>
`RUN4` twice <Return>
`RUN3` twice <Return>
`RUN2` twice <Return>
`RUN1` twice <Return>

React. The cursor jumps to the next line (M1.1)

Step 27 Enter <Ctrl>+<F> for the next search

React. The search window is opened

Step 28 Enter `M15.1` and <Return>

React. The cursor jumps to the ”Symbol” column

Step 29 Enter `AUX1` and <Return>

React. The cursor jumps to the ”Initial value” column

Step 30 Enter `0` and <Return>

React. The cursor jumps to the ”Comment” column

Step 31 Enter `Aux. marker 1` and <Return>

React. The cursor jumps to the next line (M15.2)

Step 32 Repeat Step 29 to Step 31 with the following input:

AUX2	<Return>	0	<Return>	Aux. marker 2	<Return>
AUX3	<Return>	0	<Return>	Aux. marker 3	<Return>
AUX4	<Return>	0	<Return>	Aux. marker 4	<Return>
AUX5	<Return>	0	<Return>	Aux. marker 5	<Return>
AUX6	<Return>	0	<Return>	Aux. marker 6	<Return>

React. The cursor jumps to the next line (M15.7)

Step 33 Enter <Ctrl>+<F> for the next search

React. The search window is opened

Step 34 Enter `MW1100` and <Return>

React. The cursor jumps to the ”Symbol” column

Step 35 Enter `ROTATED` and <Return>

React. The cursor jumps to the ”Initial value” column

Step 36 Enter `0` and <Return>

React. The cursor jumps to the ”Comments” column

Step 37 Enter `This word contains the rotated info` and <Return>

React. The cursor jumps to the next line (MW1101)

Step 38 Enter <Ctrl>+<F> for the next search

React. The search window is opened

Step 39 Enter `MW1200` and <Return>

React. The cursor jumps to the ”Symbol” column

Step 40 Enter `SHIFT` and <Return>

React. The cursor jumps to the ”Initial value” column

Step 41 Enter `1` and <Return>

React. The cursor jumps to the ”Comments” column

Step 42 Enter `Shift clock pulse for SFB123` and <Return>

React. The cursor jumps to the next line (MW1201)

Step 43 Enter <Ctrl>+<T> to terminate and save the SYMCOM block.

React. You can look at the Edit menu again

5.5.6 Edit Program (Blocks)

The user program is input to the AKF35 in this chapter.

The function block (FB) containing the program for the running light is first edited. The FB1 contains the declaration part in network 1, the instruction part in network 2, and network 3 with ”BE” for Block End.

5.5.6.1 Open Block Editor

Step 1 Enter `B` for ”Block”

React. A line into which you can type the block to be edited is opened.

Step 2 Enter `FB1` for block and <Return>

React. The block editor is opened and the declaration part network of the FB1 (network 1) is displayed.

5.5.6.2 Edit FB1

❒ **Edit Declaration Part of the FB (Network 1)**

The mask for the later FB1 call is defined in the declaration part. The formal operands to be used in the logic operations in the program and to be assigned actual operands when called (e.g. I..., Q...., MW... etc.) are defined here.

Step 1 Enter `RUNLI` <Return> for the name of the block.

React. A new line is opened below the line `<Identifier Type Attribute>`

Step 2 Enter `OFF` <Return>.

React. The cursor jumps to the ”Type” line

Step 3 Enter a space and <Return>

React. A window with all the possible setpoint types is opened

Step 4 Select ”BIT” with the cursor keys and confirm with <Return>

React. The cursor jumps to the Attribute column. I is already there.

Step 5 Enter a `Q` after the I.

React. A new line is opened.

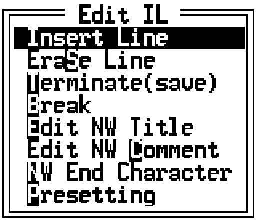
Step 6 Repeat Step 2 bis Step 5 with the following input:

LOAD	<Return>	BIT	<Return>	IQ
EN	<Return>	BIT	<Return>	IQ
SBI	<Return>	BIT	<Return>	IQ
LBI	<Return>	BIT	<Return>	IQ
ROTA	<Return>	MW	<Return>	IQ
SBO	<Return>	BIT	<Return>	OQ
LBO	<Return>	BIT	<Return>	OQ

React. The cursor now points to a new line which you must now delete.

Step 7 Confirm with <Ctrl>+<Return>

React. The processing menu is opened



Step 8 Confirm with `S` for EraSe Line

Step 9 Store the network with <Ctrl>+<T> for Terminate.

Step 10 Scroll to the next network (NW2) with <PgDn>.

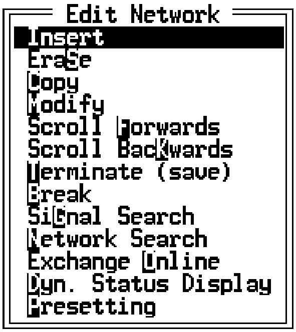
The first network of the FB is thus terminated.

❒ Insert Network

In order to edit the program in the block, an empty network must first be inserted.

Step 1 Confirm with <Ctrl>+<Return>

React. The processing menu is opened



Step 2 Enter `I` for ”Insert” (Network)

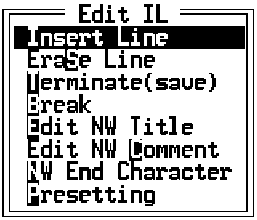
React. A new network is always inserted before the current network with the function ”Insert”. In this case, network 2 is now empty (contains only :***), ”BE” for Block End is in network 3.

❐ Edit Instruction Part of the FB (Network 2)

The instruciton part, which contains only one network in our example, contains the user program of the FB1. (Network 3 contains only :BE for ”Block End”). Function blocks are generally created in instruction list

Step 1 Confirm with <Ctrl>+<Return>

React. The processing menu is opened



Step 2 Confirm with `I` for ”Insert Line”

Step 3 Press <Return> several times in order to insert more lines

Step 4 Now enter the following lines. You can insert futher lines above the cursor during editing with <Return>.
AN <Tab> =OFF <Return>
JT =JUM1 <Return>
LD <Tab> V0 <Return>
T <Tab> =ROTA <Return>
JI <Tab> =JUM3 <Return>

Step 5 For the jump destination JUM1, move the cursor in the empty line with the cursor keys to the left to the edge of the screen:
JUM1 (to :) BC <Tab> SFB112 <Return>

Step 6 The SFB must now be parametrized. Enter the following lines:
for E: =LOAD <Return>
for AZ: AUX1 <Return>
for A: AUX2 <Return>

Step 7 Enter the following lines corresponding to Step 4:
AN <Tab> AUX2 <Return>
JT =JUM2 <Return>
BC <Tab> SFB131 <Return>

Step 8 The SFB must now be parametrized. Enter the following lines:
for BK: =SBI <Return>
for BL: =LBI <Return>
for WA: =ROTA <Return>

Step 9 Enter the following line corresponding to Step 5:
JUM2 (to :) A <Tab> PULSE_5 <Return>

Step 10 Enter the following lines corresponding to Step 4:
A <Tab> =EN <Return>
= <Tab> AUX5 <Return>
BC <Tab> SFB112 <Return>

Step 11 The SFB must now be parametrized. Enter the following lines:

for E: AUX5 <Return>
for AZ: AUX3 <Return>
for A: AUX4 <Return>

Step 12 Enter the following lines corresponding to Step 4:

AN <Tab> AUX4 <Return>
SPB =JUM4 <Return>
BC <Tab> SFB123 <Return>

Step 13 The SFB must now be parametrized. Enter the following lines:

for EF: FIXED”1” <Return>
for LR: FIXED”0” <Return>
for RS: FIXED”1” <Return>
for DW: =ROTA <Return>
for DA: SHIFT <Return>
for AB: AUX6 <Return>

Step 14 Enter the following line corresponding to Step 5:

JUM3 (to :) BC <Tab> SFB129 <Return>

Step 15 The SFB must now be parametrized. Enter the following lines:

for BK: =SB0 <Return>
for BL: =LB0 <Return>
for WE: =ROTA <Return>

Step 16 Enter the following line corresponding to Step 5:

JUM4 (to :) NOP <Return>
:***

❐ Deletion of Superfluous Empty Instruction Lines

The network is terminated with :***. Superfluous empty lines are deleted as follows:

Step 1 Move the cursor to the empty line with the cursor keys

Step 2 Enter <Ctrl>+<S> for ”EraSe Line”

❐ Terminate and save network

Step 1 Enter <Ctrl>+<T> for ”Terminate”

❐ Terminate and save block

Step 1 Enter <Ctrl>+<T> for ”Terminate”

The block FB1 is now terminated. The FB1 must now be linked into the organization block since all program control is converged here. The program is not executable without an organization block.

5.5.6.3 Edit OB1

❐ **Open OB**

Step 1 Enter B for ”Block”

React. A line is opened into which you should type the block to be edited.

Step 2 Enter OB1 for block and <Return>

React. The block editor is opened and the last network of OB1 appears with ”BE” for block end.

Step 3 Enter <Ctrl>+<Return> to open the processing menu

Step 4 Enter P for ”Presetting”

Step 5 Enter I for ”Input Mode” until IL is set.

Step 6 Enter <Esc>

Step 7 Enter <Ctrl>+<I> for ”Insert” (network)

React. A new network is always inserted in front of the current network with the function ”Insert”. In this case network 1 is now empty (contains only :***), network 2 contains ”BE” for block end.

- **Call FB in OB (unconditional, i.e. the PB is called in each scan)**

Step 1 Enter <Ctrl>+<I> for "Insert Line"

Step 2 Enter the following text for the block call:
`BC` <Tab> `FB1` <Return>

React. The following display appears:

```
     :BC   FB1
NAME :RUNLI
OFF  :     BIT?
LOAD :     BIT?
EN   :     BIT?
SBI  :     BIT?
LBI  :     BIT?
ROTA :     MW?
SBO  :     BIT?
LBO  :     BIT?
     :***
```

- **Parametrize FB in the OB1**

Step 1 The FB must now be parametrized. Enter the following lines:

for OFF:	`OFF` <Return>
for LOAD:	`LOAD` <Return>
for EN:	`EN` <Return>
for SBI:	`BIT8` <Return>
for LBI:	`BIT1` <Return>
for ROTA:	`ROTATED` <Return>
for SBO:	`RUN16` <Return>
for LBO:	`RUN1`

- **Terminate and Save Network**

Step 1 Enter <Ctrl>+<T> for "Terminate"

- **Terminate and save OB1**

Step 1 Enter <Ctrl>+<T> for "Terminate"

Step 2 Leave the "Edit" menu with <Esc>.

React. The selection bar points to the line "Edit"

The program input is herewith terminated.

❐ Look at the program in the editor

If you want to compare the program with the printout of the following page, you can select the blocks again in the editor (”Edit”, ”Block”). You can scroll in the networks with <PgDn>, <PgUp> and the cursor keys and you can compare your blocks with the following printout.
You terminate the editor with <Ctrl>+<T>.

❐ Correct possible typing errors in the program

If you made typing errors in the program, open the processing menu in the block editor with <Ctrl>+<Return>. Select the ”Modify” function. You can then select and modify the corresponding instruction with the cursor keys.
You terminate the editor by entering <Ctrl>+<T> twice.

A printout of the program which was programmed is given below.

```
C:\AKF35E\EXERCISE\OB1

NETWORK: 0001
      :BC   FB1
NAME  :RUNLI
OFF   :     OFF
LOAD  :     LOAD
EN    :     EN
SBI   :     BIT8
LBI   :     BIT1
ROTA  :     ROTATED
SBO   :     RUN16
LBO   :     RUN1
      :***


NETWORK: 0002
      :BE


C:\AKF35E\EXERCISE\FB1

NETWORK: 0001
NAME  :RUNLI
IDT   :OFF          BIT      IQ
IDT   :LOAD         BIT      IQ
IDT   :EN           BIT      IQ
IDT   :SBI          BIT      IQ
IDT   :LBI          BIT      IQ
IDT   :ROTA         MW       IQ
IDT   :SBO          BIT      OQ
IDT   :LBO          BIT      OQ
      :***


NETWORK: 0002
      :AN   =OFF
      :JT   =JUM1
      :LD   V 0
      :T    =ROTA
      :JI   =JUM3
JUM1  :BC   SFB112
NAME  :FLE
E     :     =LOAD
AZ    :     AUX1
A     :     AUX2
      :AN   AUX2
      :JT   =JUM2
      :BC   SFB131
NAME  :LBW
BK    :     =SBI
BL    :     =LBI
WA    :     =ROTA
JUM2  :A    PULSE_5
      :A    =EN
      :=    AUX5
      :BC   SFB112
NAME  :FLE
E     :     AUX5
AZ    :     AUX3
A     :     AUX4
      :AN   AUX4
      :JT   =JUM4
      :BC   SFB123
NAME  :SFW
EF    :     FIXED"1"
LR    :     FIXED"0"
RS    :     FIXED"1"
DW    :     =ROTA
DA    :      SHIFT
AB    :     AUX6
JUM3  :BC   SFB129
NAME  :LWB
BK    :     =SBO
BL    :     =LBO
WE    :     =ROTA
JUM4  :NOP
      :***
```

5.5.7 Connect PLC / Networking PLC ↔ PADT

Note The following functions in chapter 5.5.7 to chapter 5.5.12 are only possible if the PLC is properly connected.

Connect the connecting cable PADT (COM1) ↔ PLC (RS 232C).

❐ Set networking in AKF35

Step 1 Enter `T` for "SeTup"

React. The Setup menu is opened

Step 2 Enter `N` for "Networking"

React. The networking menu is opened, the selection bar points to "RS232"

Step 3 Confirm "RS232" with <Return>

React. A menu to define the transmission rate appears. The value is preset to 9 600 bits/sec

Step 4 Press <Esc> twice

Note If a program is running in your PLC, two messages are displayed. Both can be confirmed with "n" for no.

React. You can look at the Setup menu again. The networking settings are accepted: V.24 and 9 600 bits/sec. The connection is set up.

Note If a program is running in your PLC, change to "Online" with <→> and activate the function "StOp PC*"

5.5.8 First-time Parameter Assignment

The PLC is standardized and prepared for program acceptance with the following function.

Step 1 Enter S for Station

React. The actual Setup menu is opened, the selection bar points to ”PC* Station Name”

Step 2 Enter F for ”First Use of PC*”

React. The selection bar points to ”Start Function”

The line ”RAM/EPROM Version” remains RAM since the program is to be loaded into RAM (presetting).

The line ”EPROM Segment Numbers” remains unused (presetting).

Since the program is very small, only 2 segments are entered as RAM segments.

Step 3 Enter R for ”RAM Segment Numbers”

React. The bar is opened with the input width

Step 4 Enter 7,8 and <Return>

React. The input is accepted

The line ”Set up RAMZU-PADT” remains ”yes” (presetting).

The line ”Set up RAMZU-SEAB” remains ”no” (presetting).

The line ”ReServed Segment Nos.” is allocated independently by the software.

☞ **Note** The segment with the highest number which was not reserved is used for RAMZU-PADT (in our case segment 24, see below under BSW Segments).

Since the program is driven with loadable basic software, segments must be allocated for the basic software.

Step 5	Enter `W` for ”BSW Segments”
React.	The bar is opened with the input width
Step 6	Enter `19,20,21,22,23` and <Return>
React.	The input is accepted
Step 7	Enter `S` for ”Start Function”
React.	A message appears in which you are asked whether you really want to execute the function
Step 8	Enter `y` for ”yes”
React.	The first-time parameter assignment is made (message). A message appears again after termination.
Step 9	Press <Return> to confirm message
Step 10	Press <Esc> three times to terminate and leave the Setup menu.

The PLC is now ready to accept the program.

5.5.9 Link Program

The program is prepared for the PLC with the following function

Step 1 Enter L for ”Load”

React. The load menu is opened, the selection bar points to ”Program Link”

Step 2 Confirm ”Program Link” with <Return>

React. The function ”Program Link” is executed. A message appears in which the program size is defined.

Step 3 Confirm the message with <Return>

The program is now ready to be transmitted to the PLC.

5.5.10 Load Basic Software and Program to PLC

The basic software modules used, the program, the equipment list and the initial values are transmitted to the PLC with the following function.

Step 1 Enter P for ”Program to PC*”

React. The program is transmitted (messages)

Note Since the basic software for ALU 021 (and ALU 071) are also transmitted, the function is time-consuming.

The program can now be started.0

5.5.11 Start Program

Step 1 Change to ”Online” with <→>

React. The load menu is closed, the Online menu is opened; the selection bar points to ”StArt PC*”

Step 2 Confirm ”StArt PC*” with <Return>

React. A message appears asking whether you really want to start

Step 3 Confirm with y for ”yes”

React. The green ”run” LED on ALU 021 is lit on the PLC.

Step 4 Insert the simulators in the area of the inputs of the DAP 102 (avoid insertion errors!).

Step 5 Confirm the simulator switchs according to the problem statement (see page 507).

React. The program runs on DAP 102: the 8-bit pattern is displayed at Q2.1 to Q2.16.

5.5.12 Dynamic Status Display

You can now display the operands of the program during program execution.

Step 1 Enter `D` for ”Dynamic Status Display”

React. The menu is opened

Step 2 Enter `0` for ”Online Recording”

React. A menu line for block selection is opened

Step 3 Enter <blank> and <Return>

React. A selection window is opened containing all the blocks of the station

Step 4 Select FB1 with <→> and press <Return>

React. The first network of the FB1 is displayed

Step 5 Now scroll through the networks of the block with <PgDn> and <PgUp>

Step 6 You can call a processing menu with <Ctrl>+<Return>.

The fixed assignment of the steps is now terminated.

You can now take further steps as required. You can look at the status of the individual operands using the status list. You can alter signal states at short notice using the control list.

5.6 Further Exercise (Solution)

Suggestion for a further exercise: modification of the above program. The program is modified so that the bit pattern is set with auxiliary markers instead of with inputs.

- Select the OB1 and network 1 under ”Edit”, ”Blocks”
- Set ”Modify” in the processing menu
- Change the bit pattern from I2.25 to I2.32 by entering marker M12.1 instead of BIT1 and marker M12.8 instead of BIT 8
- Terminate the network and the block
- Transmit the block thus created online to the PLC without previous linkage (”Load”, ”Exchange Online”)
- Then change the control list by entering M12.1 ... M12.8 instead of Bit1 ... Bit8
- Assign the required bit pattern to the markers in the ”Binary” column
- Start control display with <F1>
- The new bit pattern is transmitted to the outputs with the 0 $\rightarrow$ 1 edge on I2.26.

5.7 Remarks about the Program Documentation

You can create the program documentation in the "Print" menu. You can output all the important data to a printout with a command file.

You can choose whether the lists should be output on the screen, in a file or on 0the printer.

The files can be processed with any ASCII editor. You are under the current station directory. You can assign the name of the file yourself when you select "Output Unit", "File".

The printer must be initialized when outputting to the printer. Initialize in the menu "SeTup", "Print".

☞ **Note** Please read the documentation about the menues in part III.

5.8 Remarks about Data Security

You can archive ("backup") or restore the whole station in the "Special" menu.

☞ **Note** Please read the documentation about the menues in part III.

Chapter 6
Task Description of AKF Station ”Example”

The software contains an example of a plant with three stations. It is used to explain the structuring possiblities with Dolog AKF. The chapter only explains the program structure and does not contain an exact program printout.
The lists of the individual programs can be output with Dolog AKF under ”Print”.

6.1 General Information

This sample plant is a sequence control system for an automatic rotary table with three stations, where each station is divided into 8 steps of a sequence control system. The emphasis in this description will be on aspects which point out the different methods of configuration.
Three identical stations of an automatic rotary table will be used to show three methods of creating structured user programs.

This program is not intended to completely describe the sequence control system of an automatic rotary table, but to give insight into the different structuring possiblities of user programs with the Dolog AKF software.

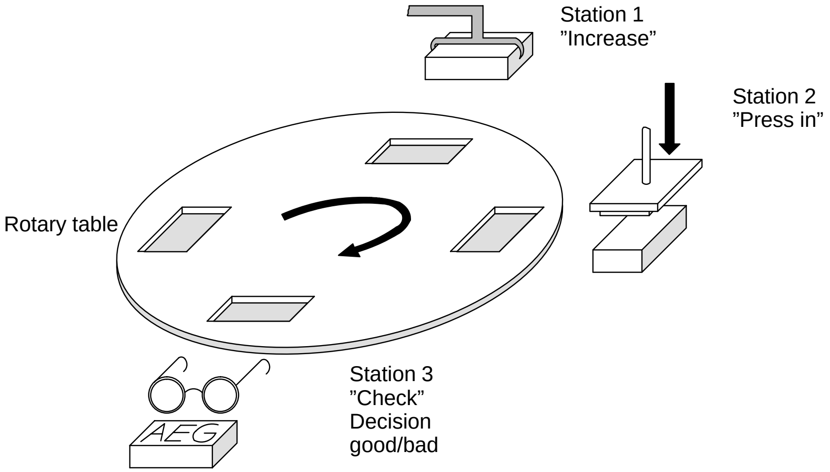


Figure 33 Diagram of an Automatic Round Table

6.2 Problem Definition

The automatic rotary table contains three stations and the rotary motion. Emphasis was placed on the stations. The operating mode selection switch was only displayed with a simplified representation of the subgroups with regard to the individual stations.
The rotary motion of the table was not handled either since it basically behaves like the functional flow of a station.
In our example we have a plant under pneumatic control partially with 100% control, where the grab is open in off-circuit state and is in the upper left limit. If the machine has the basic setting ”Upper/Left” and the operating mode selection switch is not set to ”manual” operation, the first step is started after pressing the material flow enable key.

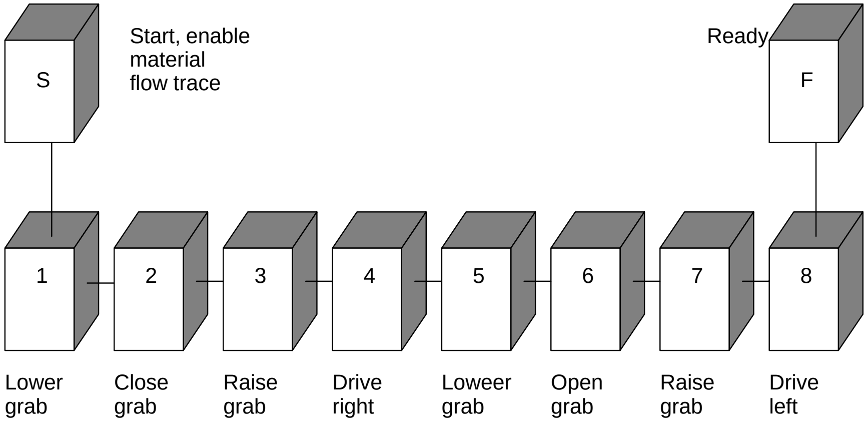


Figure 34 Sequence of Steps in a Station

If the grab is open and has the basic setting (”Upper/Left”) again, the ready signal of the relevant station is given for this flow. The ready signals are deleted after termination of the rotation.
The eight steps are the same in each station. For this reason each station is described with its own solution, whereby all stations are implemented in a single organization block.

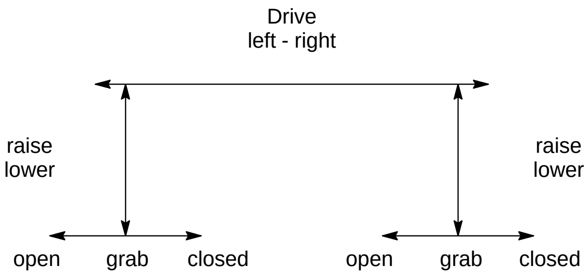


Figure 35 Motion of the Grab in a Station

Each program part describing a station has advantages and disadvantages. The degree to which the configurer takes advantage of the structuring possibilities provided depends on various criteria such as:

- the scan time
- the program length (physical memory)
- the program creation time.

The following diagrams show an operating mode switch belonging to each plant permitting each station to be operated in another operating mode.
The next superior operating mode selection switch is skipped in the following three station descriptions.

☞ **Note** The block numbers in the following diagrams correspond to the ”example” station provided on your diskettes.

6.3 Station 1

Examplel 1 describes a configuration which is equivalent to sequential processing without fine structures. This station contains only program blocks. The structure of the station is therefore the easiest to recognize. This program structure is recommended for smaller plants for which less importance is placed on speed. The configuration costs are greater than for stations 2 and 3 since user function blocks were dispensed with here. However, no formal parameters local to the FB need be declared. This configuration is not complex since everything executes sequentially.
Hwoever, this also means that all the blocks in the scan are processed and must be checked. The grab may not open, for example, when driven right/left.

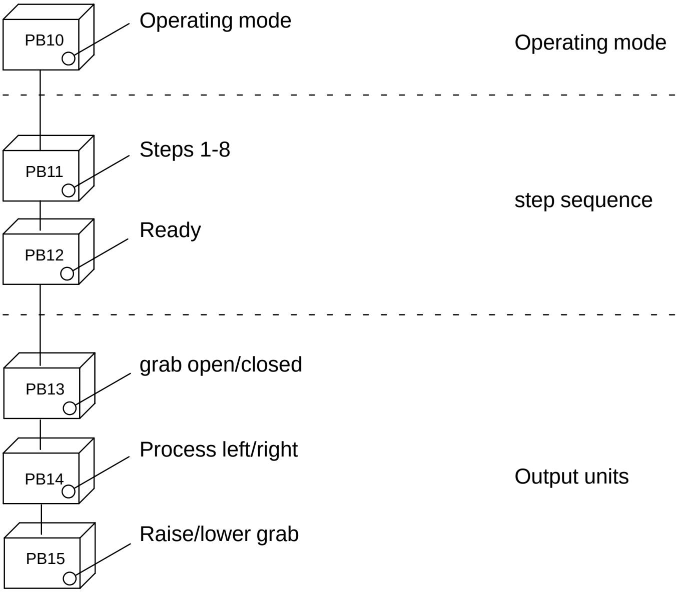


Figure 36 Representation of Station 1

6.4 Station 2

Example 2 shows sensible structuring using program and function blocks. The processing of the step sequence from steps 1 to 4 or the step sequence from 5 to 8 is event-controlled. This station provides optimal transparency with regard to configuration and maintenance.
Exmaple 2 is a configuration example for middle-sized and large plants. User function blocks which only need to be parametrized after linking into a program block are used. Smaller configuration costs and memory than in example 1 are therefore necessary because FB23, for example, is only programmed once (thus only occupies memory once) but is called three times with different parameters.
Event-controlled processing of the step sequence (steps 1-4 or steps 5-8) enables a reduction in the scan times for larger plants. Memory space is also saved by using user function blocks.
The more steps you use, the better the use of the memory space.

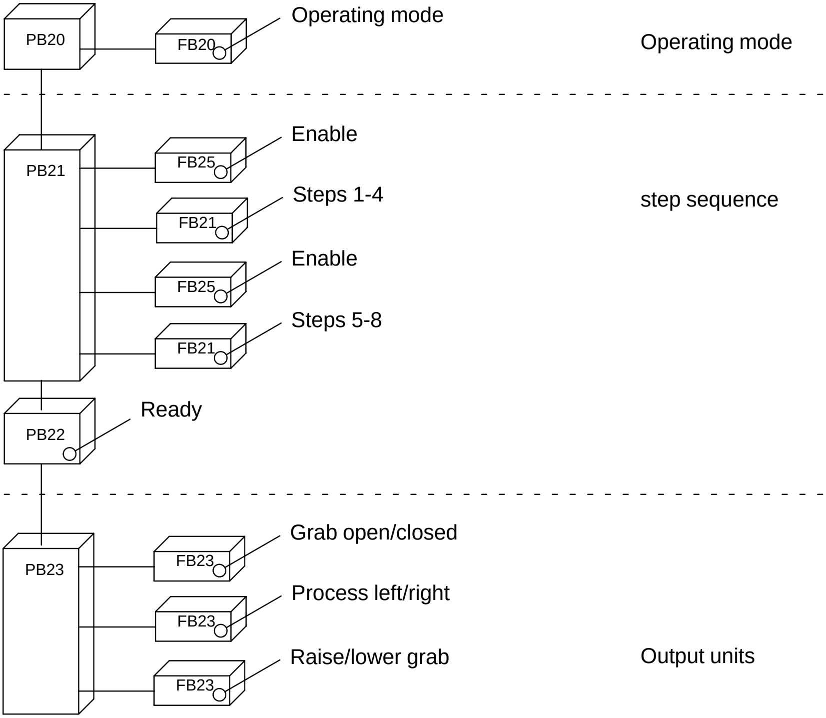


Figure 37 Representation of Station 2

6.5 Station 3

Example 3 differs from example 2 in that it has an even finer structuring. As in example 2, the step sequence is also divided into two units, whereby only one is enabled under event control. In contrast to station 2, the output unit is here divided into 6 block units. This structure only makes sense for complex automation problems in which short scan times are also of importance. Each of the 6 output units is therefore enabled under event control. The configuration costs for example 3 are therefore greater than for example 2. The enable blocks are parametrized in station 3 directly in the organization block OB1. This prevents a program block from being called. The tasks of the drive units are again shared, also saving scan time with respect to example 2. During program execution, however, parameter alterations are only possible if the OB1 can be exchanged.

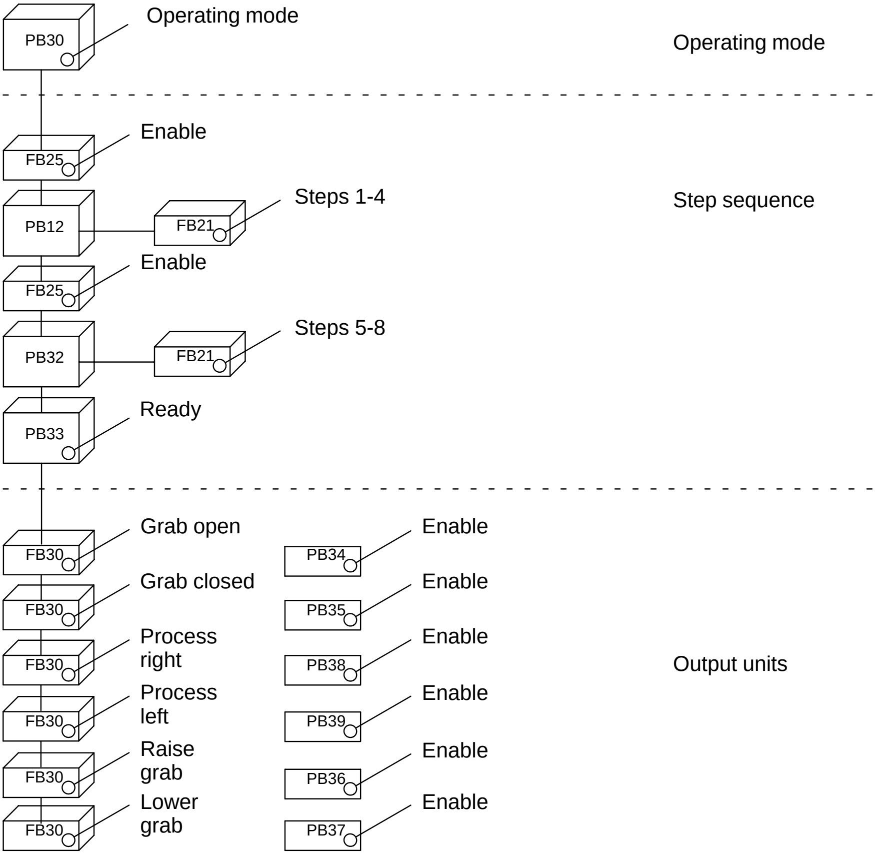


Figure 38 Representation of Station 3

AKF User Club

The AKF User Club was set up in order to provide you with optimal possibilities to save time and costs in your configuration work.

Features
All members of the User Club are provided with information about:
feature extensions,
software maintenance,
additional documentation,
applications,
courses, seminars, workshops,
current uses (universal, technologically oriented).
Joining the AKF User Club gives you the opportunity to directly exchange experience and ideas for software with other users.
You are provided with a DIN A4 file with an index for filing the corresponding information.

Membership
The members of the AKF User Club are always **known** staff of customers or of AEG departments.
Membership is confirmed by a message on the software license agreement.

☞ **Note** Please do not forget to return the software license agreement after having completely filled in your name and (company) address.

Supervision and Exchange of Applications

Address of AKF User Club:

AEG Aktiengesellschaft
Business Unit Automation
International Subdivision
Automation Technology MODICON
AKF USER CLUB, H. Herforth
Steinheimer Str. 117

D - 63500 Seligenstadt

Address of Applications Exchange:

AEG Aktiengesellschaft
Business Unit Automation
International Subdivision
Automation Technology MODICON
A91D/V12, H. Kämmerer
Steinheimer Str. 117

D - 63500 Seligenstadt

Part VI
Index

Index

A

Abort
  Data Structure: *Part III* 198
  Dynamic status display: *Part III* 239
  Equipment List: *Part III* 164
  IL-/LD-/FBD-Correction: *Part III* 122
  IL/LD/FBD: *Part III* 127
Actual operand: *Part III* 51
Address input LD: *Part III* 135
Address Mode: *Part III* 132–133
Addressing: *Part III* 132
AKF User Club: *Part V* 551
  Features: *Part V* 551
  Membership: *Part V* 551
  Supervision: *Part V* 552
ALU type: *Part III* 314
AND-block, FBD correction: *Part III* 148
Archive from memory area: *Part III* 229
Archive from station: *Part III* 283
Archive to station: *Part III* 284
Auto-repeat function: *Part III* 70

B

Backup copy: *Part II* 27
Basic sampling time
  Edit CLC: *Part III* 207
  Online CLC: *Part III* 258
Basic software editor (V. > = 6.0): *Part III* 320
Basic software version: *Part III* 186
Bit strings and back connection technique: *Part III* 132
Bit strings and front connection technique: *Part III* 132
Block editors: *Part III* 84
Block list: *Part III* 176
  Export blocks: *Part III* 308
  Import blocks: *Part III* 298
Block numbers: *Part III* 186
Block packages: *Part III* 186
Break
  Dynamic status display: *Part III* 239
  Equipment List: *Part III* 164
  IL-/LD-/FB-Correction: *Part III* 122
  IL/LD/FBD: *Part III* 127
Break Correction. *See* Break (Abort) IL/LD/FBD Correction
Bsdol operating functions: *Part III* 255
  SSN: *Part III* 255
BSW < 6.0-user
  Configure BSW (only possible with >= 6.0): *Part III* 344
  Dolog SFB address table: *Part III* 344
  Intelligent function modules (front): *Part III* 345
  Loadable BSW (only possible with >= 6.0): *Part III* 344
  POS102/POS112 (only possible with >= 6.0): *Part III* 345
BSW editor, Scroll Backwards: *Part III* 326
BSW editor, Abort: *Part III* 326
BSW editor, ALU type: *Part III* 327
BSW editor, Delete module/SFB: *Part III* 324

BSW editor, Info: *Part III* 325
BSW editor, Insert module/SFB: *Part III* 322, 324
BSW editor, Link basic software: *Part III* 328
BSW editor, Load basic software in PC*: *Part III* 328
BSW editor, Module list: *Part III* 321
BSW editor, Normalize: *Part III* 327
BSW editor, Print module list: *Part III* 324
BSW editor, Print, Module list (selection): *Part III* 325
BSW editor, Print, Print mode: *Part III* 325
BSW editor, Scroll Forwards: *Part III* 326
BSW editor, Sort: *Part III* 327
BSW editor, Terminate (Save): *Part III* 327

C

Call marker
- Control loop: *Part III* 200
- Runtime system: *Part III* 201

Call parameter /BW: *Part III* 341
Call parameter /COL: *Part III* 341
Call parameter /GR: *Part III* 341
Change data structure presetting: *Part III* 199–204
Checklist: *Part I* 6–7
- Configuration: *Part I* 6
- EPROM Operation: *Part I* 7
- Installation: *Part I* 6
- Program transmission: *Part I* 7
- Start-up: *Part I* 7
- Test: *Part I* 7

CLC: *Part III* 256–262
Command file name: *Part III* 190
Comments editor, IL: *Part III* 123
Compare
- *See also* Load, Compare
- Blocks with PC*: *Part III* 219
  - Start: *Part III* 220
- Program with PC*: *Part III* 221

Compare mode: *Part III* 220
- Byte comparison: *Part III* 220
- Fast comparison: *Part III* 220

Components for programming: *Part V* 479
Conditional block call: *Part III* 87
Configuration example: *Part I* 7
Connected PLC: *Part III* 211, 219, 236
Continuation of Parallel Line, LD: *Part III* 144
Control loop no., Online CLC: *Part III* 260
Conversion of PB to FB: *Part III* 88
Copy area, SYM/COM-Block: *Part III* 157
Copy data structure: *Part III* 198
Copy files, AKF blocks: *Part III* 291
Copy network, IL/LD/FBD: *Part III* 118
Copy of files, DOS files: *Part III* 293
Copy source (AKF blocks): *Part III* 292
Copy source (DOS files): *Part III* 293
Copy target (AKF blocks): *Part III* 292
Copy target (DOS files): *Part III* 294
Counter Down CTD (LD/FBD): *Part III* 109
Counter Up CTU (LD/FBD): *Part III* 109
Cross-reference list. *See* Print, Cross-reference list
Cross-reference mode: *Part III* 270
Ctrl + Reference characters: *Part III* 71
Cursor positioning, IL/LD/FBD: *Part III* 125

D

Data structure editor
  Attribute column: *Part III* 194
  Basic type column: *Part III* 193
  Comment column: *Part III* 195
  Display mode column: *Part III* 195
  Editor functions: *Part III* 195–199
  Number column: *Part III* 193
  Relationship between name, samples, variables: *Part III* 192
Data structure list (Export data structure): *Part III* 307
Data structures
  Data structure name: *Part III* 192, 196
  Sample: *Part III* 192
  SFBs: *Part III* 94
Delete data structure: *Part III* 197
Delete FBD element: *Part III* 135, 146
Delete LD element: *Part III* 135
Delete line in data structure: *Part III* 196
Device selection, SW-installation: *Part II* 24
DIN 19239: *Part V* 486
Directory: *Part III* 278, 313
  Station: *Part III* 278
Directory structure on PADT: *Part III* 312
Display: *Part V* 486
Display modes IL, LD, FBD. *See* Input mode
Dolog AKF main menu: *Part III* 75
Drive, Formatting: *Part III* 288
DRU 096: *Part III* 337
DRU 120: *Part III* 337
DRU 1200: *Part III* 338
DRU 292: *Part III* 337
DRU 293: *Part III* 338
Dynamic status display: *Part III* 130, 237–250
  Break: *Part III* 239
  Display mode: *Part III* 238
  Display NW comment: *Part III* 241
  Display Status list: *Part III* 244
  MV Mode: *Part III* 240
  Online recording: *Part III* 238
  Presetting: *Part III* 241
  Single shot processing: *Part III* 242
    Search block: *Part III* 244
    Start single shot: *Part III* 242
  Single shot start: *Part III* 243
  Triggered recording: *Part III* 245, 246
    Display status list: *Part III* 247
    Examples: *Part III* 249
    Next recording: *Part III* 246
    Previous recording: *Part III* 247
    Select recording: *Part III* 247
    Start display: *Part III* 245
    Trigger conditions: *Part III* 248

E

Edit: *Part III* 38, 78; *Part V* 480
  Block: *Part III* 38
  Closed-loop control: *Part III* 38
  Command file: *Part III* 38
  Compile blocks: *Part III* 38
  Data structures: *Part III* 38
  Dolog-SFB address list: *Part III* 38
  Equipment list: *Part III* 38
  Overview: *Part III* 38, 79–82
  Replace signals: *Part III* 38
  Symbols and comments: *Part III* 38
  Title block: *Part III* 38
  Translate data/blocks: *Part III* 38
Edit command file: *Part III* 188
Edit comment, Edit CLC: *Part III* 210
Edit comment, Online CLC: *Part III* 262
Edit equipment list, Example: *Part III* 170
Edit, Blocks: *Part III* 83–133
Edit, Blocks, Start input: *Part III* 114
Edit, Command file: *Part III* 188–190
Edit, Compile (Blocks): *Part III* 178

Edit, Dolog SFB address list (BSW < V6.0), Conditions: *Part III* 184–187
Edit, Dolog SFB address list (BSW less than V6.0): *Part III* 184–187
Edit, Dolog-SFB address list: *Part III* 186
Edit, Equipment list: *Part III* 162–170
Edit, Replace signals: *Part III* 174–177
Edit, Replace signals, Start: *Part III* 175
Edit, Start translation (SYM/COM block): *Part III* 179
Edit, Structure editor: *Part III* 191–199
Edit, Symbols and comments: *Part III* 152–161
Edit, Title block: *Part III* 171–173
Edit, Title block, Start: *Part III* 171
Edit, Translate (SYM/COM blocks): *Part III* 179–183
Editing in different display modes: *Part III* 115
  See also IL-Editor, LD-Editor, FBD-Editor
EPROM segment no., First use of PC*: *Part III* 318
Epson EX-800: *Part III* 338
Epson FX-80: *Part III* 339
Equal, IL: *Part III* 101
Equipment list column 1: *Part III* 163
Equipment list column 2: *Part III* 163–166
Equipment list column 3: *Part III* 166
Equipment list column 4: *Part III* 167
Equipment list column 5: *Part III* 167
Equipment list column 6: *Part III* 167–169
Equipment list column 7: *Part III* 169
Equipment list comments: *Part III* 166
Equipment list front connection: *Part III* 164
Equipment list rear connection: *Part III* 165
Erase
  AKF blocks: *Part III* 290
  DOS files: *Part III* 290
Erase area, SYM/COM-Block: *Part III* 158
Erase field/line, SYM/COM-Block: *Part III* 156
Erase line, IL correction: *Part III* 121
Erase network, FBD correction: *Part III* 151
Erase network, IL/LD/FBD: *Part III* 117
Erase network, LD correction: *Part III* 141
Example for beginners (AKF35): *Part V* 505–540
Exchange Online, FBs (Change in declaration part): *Part III* 215
Exchange Online, IL/LD/FBD: *Part III* 129
EXIT. *See* MS-DOS
Export blocks: *Part III* 307
  To DOS file: *Part III* 308
  To export file: *Part III* 307
Export data structures: *Part III* 305
Export file name (export): *Part III* 306
Export symbols and comments: *Part III* 308
Extended pulse TEP, IL: *Part III* 100

F

FBD address input: *Part III* 146
FBD editor: *Part III* 145
FBD editor, Correction mode: *Part III* 147, 149, 150, 151
FBD elements: *Part III* 106, 107, 109, 110, 113
  LD correction: *Part III* 141
FBD symbols: *Part III* 107
File output, Output unit: *Part III* 219, 265

First Programming Steps with AKF35: *Part V* 505–540
First use of PC*: *Part III* 236, 316, 332, 333
  Additional functions for ALU 011, ALU 061: *Part III* 320–342
  BSW segments >= V.6.0: *Part III* 320
  BSW-Configuration, BSW>=V6.0: *Part III* 320
  Start: *Part III* 317
  Valid for all ALU types: *Part III* 316–319
Flipflops: *Part III* 108
Form feed: *Part III* 340
Formal operand: *Part III* 51
Function block diagram FBD: *Part III* 42, 43, 85, 107, 110, 111, 265; *Part V* 483, 490, 492, 493, 496
Function block FB: *Part III* 42, 88; *Part V* 483, 488, 492, 493, 496, 497, 498
  Actual operands: *Part V* 497
  Call: *Part III* 92
  Declaration part: *Part III* 89; *Part V* 497, 498
  Edit: *Part III* 89
  Edit declaration part: *Part III* 90
  Exchange online after change in declaration part: *Part III* 215
  Formal operands: *Part V* 497
  Instruction part: *Part III* 89, 92; *Part V* 497
Function keys: *Part III* 66

G

Global cross-reference list: *Part III* 269
Global mode, Activate: *Part III* 205
Global mode, Normalize: *Part III* 206
Global mode, Online CLC: *Part III* 204, 257
Global mode, Passivate: *Part III* 205–206
Graphic characters. *See* Key table
Greater than, IL: *Part III* 101
Greater than/equal, IL: *Part III* 101

H

Hardcopy. *See* Screen copy
Hardware requirements
  A350: *Part III* 331
  A500: *Part III* 331
Help function (helptext): *Part III* 77
Horizontal spread
  FBD correction: *Part III* 150
  LD correction: *Part III* 137

I

IL editor: *Part III* 115, 116, 118
  Correction mode: *Part III* 119, 121, 122, 123, 124
Import data structures: *Part III* 295
Import file name (import): *Part III* 297
Importing blocks: *Part III* 297
  From DOS file: *Part III* 299
  From export file: *Part III* 298
  From station: *Part III* 298
Importing symbols and comments: *Part III* 299
Independent of slot reference. *See* Programming independent of slot reference
Initial start of PC*: *Part III* 236
Initial start-up of the PC*: *Part III* 255
Initial values: *Part III* 96, 152, 218
Initial values for signals: *Part III* 154
Input mode: *Part III* 131
  Display modes IL, LD, FBD: *Part III* 95
Input symbols and comments: *Part III* 153
Input, FBD Correction: *Part III* 149
Insert comment line, SYM/COM-Block: *Part III* 161
Insert FBD element: *Part III* 135, 146
Insert LD element: *Part III* 135

Insert line in data structure: *Part III* 196
Insert line, IL correction: *Part III* 120
  Example: *Part III* 120
Insert network, IL/LD/FBD: *Part III* 117
Installation: *Part II* 29
Instruction list IL: *Part III* 42, 85, 95, 265; *Part V* 483, 487–488, 492, 493, 496, 497
  Structure: *Part V* 487

J

Jumps, IL. *See* Rules for IL structure

K

Key input memory. *See* Learn
Key macro: *Part III* 67, 70
Key table: *Part III* 72
  +: *Part III* 128
  -: *Part III* 128
  Alt: *Part III* 172
  Alt+character: *Part III* 69, 70
  Alt+F2: *Part III* 68
  Alt+F8: *Part III* 69
  Alt+F9: *Part III* 70
  Alt+number: *Part III* 66
  Backspace: *Part III* 67
  backspace: *Part III* 154, 172
  backtab: *Part III* 119, 123, 154, 172, 190
  Bild: *Part III* 65
  Characters: *Part III* 80
  Ctrl: *Part III* 65
  Ctrl + B: *Part III* 172
  Ctrl + C: *Part III* 283, 285, 290
  Ctrl + cursor right: *Part III* 148
  Ctrl + D: *Part III* 123, 190
  Ctrl + G: *Part III* 123, 190
  Ctrl + N: *Part III* 190
  Ctrl + PgDn: *Part III* 137, 148, 154
  Ctrl + PgUp: *Part III* 148, 154
  Ctrl + reference character: *Part III* 239, 243, 246
  Ctrl + Return: *Part III* 189
  Ctrl + S: *Part III* 123, 190
  Ctrl + T: *Part III* 190
  Ctrl + Y: *Part III* 190
  Ctrl+A: *Part III* 67
  Ctrl+C: *Part III* 72
  Ctrl+D: *Part III* 67
  Ctrl+F: *Part III* 67
  Ctrl+M: *Part III* 67
  Ctrl+R: *Part III* 67
  Ctrl+S: *Part III* 67
  Ctrl+Y: *Part III* 67
  Cursor keys: *Part III* 66, 67, 71, 72, 77, 119, 123, 135, 137, 146, 148, 154, 172, 190
  Del: *Part III* 65, 67, 119, 123, 137, 148, 154, 172, 190
  Druck: *Part III* 65
  Einfg: *Part III* 65
  Eing Losch: *Part III* 65
  End: *Part III* 65, 67, 123, 154, 190
  Ende: *Part III* 65
  Esc: *Part III* 65, 67, 72, 77, 116, 119, 123, 130, 135, 137, 146, 148, 154, 190, 239, 243, 246, 251, 253, 254
  F1: *Part III* 77, 251, 253, 254
  F10: *Part III* 77
  F2: *Part III* 77, 254
  F9: *Part III* 255
  Graphic characters: *Part III* 66, 172
  Home: *Part III* 65, 67, 123, 154, 190
  Ins: *Part III* 65, 67, 72, 119, 123, 154, 172, 190
  Letters: *Part III* 137, 148
  Lösch oder Entf: *Part III* 65
  Mouse: *Part III* 334
  Number block: *Part III* 172
  PgDn: *Part III* 65, 77, 116, 135, 137, 146, 148, 154, 172, 190, 239, 243
  PgUp: *Part III* 65, 77, 116, 135, 137, 146, 148, 154, 172, 190, 239, 243

Pos1: *Part III* 65
PrtSc: *Part III* 65
Reference characters: *Part III* 66
Return: *Part III* 67, 71, 72, 77, 80, 130, 135, 137, 146, 148, 172, 313
Strg: *Part III* 65
tab: *Part III* 119, 154
Keyboard operation: *Part III* 65–72

L

Ladder diagram LD: *Part III* 42, 43, 85, 105, 106, 265; *Part V* 483, 489, 492, 493, 496
LD editor: *Part III* 134, 135, 146
LD editor, Correction mode: *Part III* 136, 137, 138, 139, 148
LD symbols: *Part III* 105
Learn: *Part III* 69
Less than, IL: *Part III* 101
Less than/equal, IL: *Part III* 101
Line comment, IL: *Part III* 116
Line editor: *Part III* 67
Link mode: *Part III* 315
Complete Retranslation (Representation): *Part III* 315
Without comments: *Part III* 315
Without Retranslation: *Part III* 316
Load: *Part III* 38, 211–235; *Part V* 480
Archive memory area: *Part III* 38
Compare: *Part III* 38
EPROM-Handling: *Part III* 38
Exchange Online: *Part III* 38
Initial values to PC*: *Part III* 38
Online Exchange: *Part III* 38
Optimizing of PC* memory: *Part III* 38
Program Link: *Part III* 38
Program to PC*: *Part III* 38
Read equipment list: *Part III* 38
Read out PC*: *Part III* 38
Restore memory area: *Part III* 38
Set date/time: *Part III* 38
Signal memory PADT - PC*: *Part III* 38
Signal memory PC* - PADT: *Part III* 38
Load block list: *Part III* 177
Load data structure: *Part III* 197
Load, Archive memory area: *Part III* 228–229
Load, Compare: *Part III* 219–221
Load, Initial values to PLC: *Part III* 226–227
Load, Link program: *Part III* 212
Load, Online exchange: *Part III* 214–216
Load, Optimizing PLC Memory: *Part III* 224
Load, Program to PC*: *Part III* 212–213
Load, Read equipment list: *Part III* 217
Load, Read out PC*: *Part III* 218
Load, Read out PC*, Start: *Part III* 218
Load, Restore memory area: *Part III* 230–231
Load, Set date/time: *Part III* 222–223
Load, Signal memory PADT to PLC: *Part III* 225
Load, Signal memory PC* to PADT: *Part III* 224–225
Local cross-reference list: *Part III* 269
Local mode, Online CLC: *Part III* 209, 260
Logical slot reference: *Part III* 169

M

Mask for directory
AKF blocks: *Part III* 279
DOS files: *Part III* 280
Max. number of blocks: *Part III* 314
Max. RVL-runtime, Online CLC: *Part III* 259

Memory area (source) from which to restore: *Part III* 231
Memory area (target) into which to restore: *Part III* 231
Menu
Dolog AKF main menu: *Part III* 74, 75
Edit: *Part III* 74, 78
Load: *Part III* 74
Online: *Part III* 74
Print: *Part III* 74
Setup: *Part III* 74
Special: *Part III* 74
Mixed Equipment, Front Connection / Rear Connection: *Part III* 170
Modify network, IL/LD/FBD: *Part III* 118
Modnet 1/SFB
Configuration requirements: *Part III* 332
Networking: *Part III* 332
Modnet 2/NP
Configuration requirements: *Part III* 333
Networking: *Part III* 333
Monitor output, output unit: *Part III* 265
Mouse: *Part III* 64
MS-DOS: *Part III* 40
MS-DOS: *Part III* 287, 288, 294
From MS-DOS back to AKF (EXIT): *Part III* 309
MS-DOS Operating System. *See* MS-DOS
Multiple output assignment: *Part III* 139

N

NEC P60/P70. *See* PRT 294, PRT 295
Negate input, FBD correction: *Part III* 149
Nesting depth: *Part III* 42, 43; *Part V* 497
Nesting function blocks: *Part III* 93
Network comment, IL/LD/FBD correction: *Part III* 122, 123
Network Scroll Backwards, IL/LD/FBD: *Part III* 126
Network scroll forwards, IL/LD/FBD: *Part III* 126
Network Search Signal, IL/LD/FBD: *Part III* 128
Network Search, IL/LD/FBD: *Part III* 129
No networking: *Part III* 334
Node number SFBs (TN): *Part III* 169
NOP instruction, Rules for IL structure: *Part III* 104
Normalize equipment list: *Part III* 165
Normally closed contact in parallel line, LD: *Part III* 143
Normally closed contact, LD correction: *Part III* 139
Parallel ladder diagram line: *Part III* 139
Serial ladder diagram line: *Part III* 139
Normally open contact in parallel line, LD: *Part III* 142
Normally open contact, LD correction: *Part III* 138
Not equal, IL: *Part III* 101
Numerical block: *Part III* 66
NW end character, IL correction: *Part III* 124

O

Occupancy mode: *Part III* 272
Old or new signal: *Part III* 175
Online: *Part III* 39, 236–262; *Part V* 480
Closed-loop control: *Part III* 39
Control list: *Part III* 39
Dynamic status display: *Part III* 39
Force list: *Part III* 39
Start PC*: *Part III* 39
Start PLC: *Part III* 236

Status list: *Part III* 39
Stop PC*: *Part III* 39
Terminal Mode: *Part III* 39
Online Exchange, IL/LD/FBD: *Part III* 129
Online, Closed loop control: *Part III* 256–262
Online, Control list: *Part III* 252–253
Online, Dynamic status display: *Part III* 237–250
Online, Force list: *Part III* 254
Online, Status list: *Part III* 251
Online, Stop PLC: *Part III* 237
Online, Terminal Mode: *Part III* 333–334
Online, Terminal mode: *Part III* 240–250, 255, 332–334
Operand table: *Part III* 51
Operand value ranges: *Part III* 44
Operands: *Part V* 487
Operating functions, START: *Part III* 236
Operations: *Part III* 52; *Part V* 487
Operations and operand types: *Part III* 52
Optionally configurable V.24 interface: *Part III* 336
OR-block, FBD correction: *Part III* 150
Organization block OB: *Part III* 42; *Part V* 488, 489, 490, 492, 493, 496, 498
Edit: *Part III* 85
Edit (closed-loop control): *Part III* 86
Output mode: *Part III* 265
Directory
AKF blocks: *Part III* 279
DOS files: *Part III* 281
Output monitoring: *Part III* 125
Output unit: *Part III* 183, 335
Output, LD correction: *Part III* 139
Existing network: *Part III* 139
New network: *Part III* 139
Overview
Edit block: *Part III* 81
Page: *Part III* 81
Program structure: *Part III* 80
Search for block: *Part III* 81

P

PADT: *Part III* 66, 70, 76, 319
Page mode: *Part III* 267
Parallel line, LD: *Part III* 142
Parameters for formatting: *Part III* 288
Parameters to copy (DOS files): *Part III* 294
Path, Explanation: *Part III* 74
PB networks: *Part III* 42
PB-/FB-/SFB-conditional call, LD/FBD: *Part III* 140
PB-/FB-/SFB-unconditional call, LD/FBD: *Part III* 140
PC* station name: *Part III* 313
Phase, Online CLC: *Part III* 208, 260
POS.ASD: *Part III* 94
Preparation for configuration
Chapters of PLC user manuals: *Part I* 4
Checklists: *Part I* 6
Preparations in AKF for COMAKF: *Part I* 17
Programming sequence: *Part I* 8
Presetting
Dynamic status display: *Part III* 241
IL/LD/FBD: *Part III* 131
IL/LD/FBD Correction: *Part III* 124
Print: *Part III* 39, 263–276; *Part V* 480
Command file: *Part III* 39
Cross-reference list: *Part III* 39
Data structures: *Part III* 39
Equipment list: *Part III* 39
Overview: *Part III* 39
Program log: *Part III* 39
Signal memory contents: *Part III* 39
Signal occupancy list: *Part III* 39
Station Setup: *Part III* 39
Symbols and comments: *Part III* 39

Print lines/page: *Part III* 340
Print prog. log with symbols and comments: *Part III* 266
Print signal list: *Part III* 225
Print start page number: *Part III* 264
Print title block: *Part III* 181
Print, Command file: *Part III* 263, 274
Print, Cross-reference list: *Part III* 269–270
Print, Data structure: *Part III* 276
Print, Data structures: *Part III* 263
Print, Equipment list: *Part III* 263, 268
Print, Overview: *Part III* 263–264
Print, Prog. log, local cross-reference list: *Part III* 266
Print, Program log: *Part III* 265–267
Print, Setup station: *Part III* 263
Print, Signal memory contents: *Part III* 263, 275
Print, Signal occupancy list: *Part III* 271–272
Print, start print: *Part III* 264
Print, Station Setup: *Part III* 273
Print, Symbols and comments: *Part III* 263, 268
Printer output, output unit: *Part III* 219, 265
Printer selection: *Part III* 336
Program block PB: *Part III* 42, 88; *Part V* 488, 489, 490, 492, 493, 496, 498
  Edit: *Part III* 87
Program EPROM (with EPS, not with AKF): *Part III* 232–235
Program flowchart. *See* Programming sequence
Program Log. *See* Print, Prgram Log
Program overview. *See* Print Overview
Programming example: *Part V* 541–550
Programming independent of slot reference: *Part III* 167
Programming on the bus: *Part III* 332, 333
Programming Sequence: *Part I* 8
PRT 294, PRT 295: *Part III* 339
Pulldown menues: *Part III* 76
  Archivation: *Part III* 282
  Archive memory area: *Part III* 228
  Colors Help: *Part III* 341
  Colors messages: *Part III* 341
  Colors pulldown: *Part III* 341
  Command editor: *Part III* 188, 189
  Compare blocks: *Part III* 219
  Compile blocks: *Part III* 178
  Copy diskette: *Part III* 289
  Copy files (DOS): *Part III* 293
  Copy network: *Part III* 118
  Copy of files (AKF): *Part III* 291
  Data structure presetting (alter attribute): *Part III* 199
  Data struture name: *Part III* 196
  Directory: *Part III* 278
  Display single shot: *Part III* 243
  DRU 096: *Part III* 337
  DRU 120: *Part III* 337
  DRU 292: *Part III* 337
  DRU 293: *Part III* 338
  Dynamic status display: *Part III* 130, 237, 239
  Edit data structures: *Part III* 195
  Edit FBD: *Part III* 147
  Edit IL: *Part III* 119
  Edit LD: *Part III* 136
  Edit symbols and comments: *Part III* 155
  Epson EX-800: *Part III* 338
  Epson FX-80: *Part III* 339
  Export: *Part III* 305
  Export, Blocks: *Part III* 307, 308
  Export, Data structures: *Part III* 305
  Export, Symbols and Comments: *Part III* 308
  FBD elements: *Part III* 147
  First use of PC*: *Part III* 316
  Formatting: *Part III* 287
  Freely config. printer RS232: *Part III* 336
  Import: *Part III* 295

Import, Blocks: *Part III* 297, 298, 299
Import, Data structures: *Part III* 295
Importing symbols and comments: *Part III* 299
Link mode: *Part III* 315
Load: *Part III* 211
Networking PC* and PADT: *Part III* 331
Online: *Part III* 236
Online CLC, Global mode: *Part III* 204, 257
Online, CLC: *Part III* 204, 256
Presetting: *Part III* 131
Print: *Part III* 263
Print command file: *Part III* 274
Print cross-reference list: *Part III* 269
Print data structures: *Part III* 276
Print equipment list: *Part III* 268
Print program log: *Part III* 265
Print signal allocation list: *Part III* 271
Print signal memory contents: *Part III* 275
Print survey: *Part III* 263
Print symbols and comments: *Part III* 268
Print system information: *Part III* 309
Printer default values: *Part III* 183
Printer initialization: *Part III* 335
Printer selection: *Part III* 336
Process network: *Part III* 116, 134, 145
Read out PC*: *Part III* 218
Replace signals: *Part III* 174
Restore: *Part III* 285
Restore memory area: *Part III* 230
Set date/time: *Part III* 222
Setup: *Part III* 311
Setup station: *Part III* 273
Special: *Part III* 277
Station: *Part III* 313
SYKON: *Part III* 330
Table of contents (DOS): *Part III* 280
Text memory editor: *Part III* 159
Title block: *Part III* 171
Translate data block: *Part III* 179
Trigger conditions: *Part III* 248
Triggered recording: *Part III* 245, 246
Variable basic types for data structure: *Part III* 193
Pulse Extended TEP (LD/FBD): *Part III* 110
Pulse TP (LD/FBD): *Part III* 110
Pulse TP, IL: *Part III* 100

R

RAM segment no., First use of PC*: *Part III* 318
RAM-/EPROM-version, First use of PC*: *Part III* 318
Recursive call: *Part III* 43; *Part V* 498
Reduction, Online CLC: *Part III* 208, 260
Reference characters: *Part III* 65, 71, 72, 74
REGELN.ASD: *Part III* 94
Relocate area, SYM/COM-Block: *Part III* 158
Replace text, SYM/COM-Block: *Part III* 161
Replacement characters: *Part III* 293
Reserved segment numbers, First Use of PC*: *Part III* 319
Restore to station: *Part III* 286
Restoring from station: *Part III* 286
Return of parallel line, LD: *Part III* 143
RS 232C networking: *Part III* 331
RS memory, LD/FBD: *Part III* 108
Rule for bit string processing: *Part III* 132
Rules for IL structure: *Part III* 96
Arithmetic: *Part III* 102
Bracketed operations: *Part III* 97
Comparator: *Part III* 101

Counters: *Part III* 99
Dolog AKF with Dolog B: *Part III* 352
Dolog B with Dolog AKF: *Part III* 352
Formal operands in FBs: *Part III* 103
Initializing pulse: *Part III* 96
Jumps: *Part III* 104
Logic operations with constants: *Part III* 98
Logic sequence: *Part III* 97
Marker bit: *Part III* 96
Memory operations: *Part III* 98
Mixed logic operations: *Part III* 98
Network start/end: *Part III* 96
NOP instruction: *Part III* 104
Standardizing operations: *Part III* 97
Timers: *Part III* 100
Transfer: *Part III* 103
Runtime exceeded, Online CLC: *Part III* 259
Runtime of user programs: *Part III* 60
Runtime system, CLC: *Part III* 256–262

S

Sampling time, Online CLC: *Part III* 210, 261
Screen copy: *Part III* 68
Screen output, Output unit: *Part III* 219
Screen status line: *Part III* 76
Screensave. *See* Screen copy
Search function, SYMCOM: *Part III* 155
Search operand: *Part III* 156
Search Signal. *See* Network Search Signal, IL/LD/FBD
Search text, SYM/COM-Block: *Part III* 160
Segment number: *Part III* 186
Set date: *Part III* 222
Set time: *Part III* 223
Set up PLC (date/time): *Part III* 223
Set up RAMZU-PADT: *Part III* 319
Set up RAMZU-SEAB: *Part III* 319
Setup: *Part III* 40, 311–342; *Part V* 480
Colors: *Part III* 40
Networking: *Part III* 40
Print: *Part III* 40
Station: *Part III* 40
SYKON: *Part III* 330, 332–334
System: *Part III* 40
Setup, Colors: *Part III* 341–342
Setup, First use of PC*, Start: *Part III* 317
Setup, Networking: *Part III* 331
Setup, PC* Station: *Part III* 313–330
Setup, Plant: *Part III* 312
Setup, Print: *Part III* 335
SFB. *See* Standard function block SFB
Signal occupancy list. *See* Print , Signal occupancy list
Signal occupancy list, Example: *Part III* 272
Signal table
Control list: *Part III* 253
Force list: *Part III* 254
Status list: *Part III* 251
Single shot processing, Dynamic status display: *Part III* 242
Software
Call: *Part II* 30
Installation: *Part II* 29
Source directory (import): *Part III* 296
Source or target drive: *Part III* 289
Special: *Part III* 40, 277–310; *Part V* 480
Archive the station: *Part III* 40
Copy disk complete: *Part III* 40
Copy files: *Part III* 40
Directory: *Part III* 40
End of station handling: *Part III* 40
Erase a station: *Part III* 40

Erase files: *Part III* 40
Export: *Part III* 40
Format disks: *Part III* 40
Import: *Part III* 40
MS-DOS Operating system: *Part III* 40
Restore a station: *Part III* 40
System information: *Part III* 40
Special keys: *Part III* 71
Special, Archive of Station: *Part III* 282–284
Special, Archive station, Start: *Part III* 283
Special, copy complete diskette, start: *Part III* 289
Special, Copy Disk Complete: *Part III* 289
Special, Copy files: *Part III* 291–294
Special, Copy of files, Start (AKF/DOS): *Part III* 291
Special, Directory: *Part III* 278–281
AKF blocks: *Part III* 278
DOS files: *Part III* 280
Special, End of station processing: *Part III* 310
Special, Erase a Station: *Part III* 287
Special, Erase File: *Part III* 290
Special, Export: *Part III* 305–308
Special, Format disks: *Part III* 287–288
Special, Formatting, Start: *Part III* 288
Special, Import: *Part III* 295–304
Special, Operating System MS-DOS: *Part III* 309
Special, Restore station: *Part III* 285–286
Special, Restore station, Start: *Part III* 285
Special, System information: *Part III* 309
Special, Table of contents, Start: *Part III* 278
SR memory, LD/FBD: *Part III* 108
Standard address list: *Part III* 184
Standard function block library. *See* STDFB library
Standard function block SFB: *Part III* 42, 94; *Part V* 492, 499
Actual operand: *Part V* 499
Data structures: *Part V* 499
Declaration part: *Part V* 499
Formal operand: *Part IV* 358
Instruction part: *Part V* 499
TN: *Part III* 169
Start (Export): *Part III* 306
Start (Import): *Part III* 296
Start archiving memory area: *Part III* 228
Start page number: *Part III* 264
Start restoring memory area: *Part III* 230
Status CL, Online CLC: *Part III* 261
Status control loop, Online CLC: *Part III* 209
Status line: *Part III* 76
Status marker control loop: *Part III* 209
Status marker runtime system: *Part III* 202
STDFB library: *Part III* 140
Store block list: *Part III* 177
Store data structure: *Part III* 197
Structure of instruction line: *Part V* 487
Structured Programming: *Part V* 481
Summary of contacts: *Part III* 45
Summary of markers (bit): *Part III* 46
Summary of MW, MD, MF: *Part III* 47
Switch-off delay
TOF (LD/FBD): *Part III* 111
TOF, IL: *Part III* 100
Switch-on delay
Stored TS (LD/FBD): *Part III* 111
Stored TS, IL: *Part III* 100
TON (LD/FBD): *Part III* 111
TON, IL: *Part III* 100
SYM/COM block: *Part III* 152; *Part V* 492, 503

SYM/COM editor: *Part III* 153
Symbols for signals: *Part III* 154
SYRES: *Part III* 329–330, 332–334
System backup. *See* SYKON, SYRES, Operating functions
System marker: *Part III* 54
  Definition: *Part III* 54
  Marker (bit): *Part III* 55
  Marker words: *Part III* 57

T

Target directory (export): *Part III* 306
Terminal mode: *Part III* 236
Terminal operation. *See* Online, Terminal Mode
Terminate
  Edit CLC: *Part III* 210
  IL/LD/FBD correction: *Part III* 121
  Overview: *Part III* 82
Terminate (Save)
  Data structure: *Part III* 198
  Equipment List: *Part III* 164
  Online CLC: *Part III* 262
Terminate (save), IL/LD/FBD: *Part III* 126
Terminate correction. *See* Terminate, IL/LD/FBD correction
Text memory editor, SYM/COM-Block: *Part III* 159
Time base of counters: *Part III* 60
Time management table (ZVT). *See* ZVT
Time-critical processes: *Part V* 482
Title block editor: *Part III* 171
Title Block File: *Part III* 182
Title Block Height: *Part III* 173
Title block name: *Part III* 173
Title block width: *Part III* 173
TITLE.DOC: *Part III* 181
TN, Node number SFBs: *Part III* 169
Toggle: *Part III* 72
Translate (SYM/COM block), Changes: *Part III* 180
Translate (SYM/COM block), Delete DB0-DB9: *Part III* 180
Transmission rate: *Part III* 331
Trigger conditions: *Part III* 248
Triggered Recording, Dynamic status display: *Part III* 245
Triggered status display, Examples: *Part III* 249
Typewriter keyboard: *Part III* 65, 71

U

Unconditional block call: *Part III* 87

V

Validity ranges AKF12: *Part III* 42
Vertical spread
  FBD correction: *Part III* 151
  LD correction: *Part III* 138

W

Wildcards: *Part III* 293

Z

ZVT: *Part III* 256

Part VII
Appendix

Notes to AKF35EN, Version 6.4
Programming Instructions No. DOK-276529.22

I Notes to ”PADT on Modnet 2/NP” together with AKF35

Warning **If you drive your PADT on Modnet 2/NP, the PADT can crash in the following cases:**

– **when attempting to include a Modent 2/NP networking if the RS 232 connection is still active at the same time.**
  Remedy:
  Set the type of networking to Networking = None in the menu ”Setup → Networking” and remove the connections PADT ↔ PLC before starting the Modnet 2/NP networking**.**

– **when releasing an active Modnet 2/NP networking (i.e. Networking = None)**
  Remedy:
  No release of the networking or release of networking after terminating work on your PADT.
  In this case you must reboot the PADT if you want to continue work.

Note You will find some information about working with the software AKF35EN, Version 6.4 in files with the extension .DOC or .TXT on the second AKF35EN system disk.

II Modicon A350/A500 with BSW <V5.06 and <V6.01 together with AKF35 V6.x: Bsdol function RAMZU-SEAB

When programming or operating your PLC with AKF35 on Modnet 2/NP please note the following:

Caution If you have carried out the first use of PLC with AKF35 V6.x you may not use for other purposes the free segment area which is located in front of the area used from RAMZU-SEAB. Otherwise you risk a failure of the perfomance ”AKF on bus”.
(With AKF35 V5.x generally a complete segment is used for RAMZU-SEAB, that means no change is necessary.)

Note This failure does not occur any more with basic software version 5.06 or 6.01.

Remedy:
If you need this free area urgently, you have to redefine the area for RAMZU-SEAB online on the first address of this segment. The now following free area can be used for other purposes.

B Belongs to software kit E-No. 424-271510

For this proceed as the following (entries are underlined):

Step 1 ”Online” → ”Terminal Mode”

Reaction On the screen appears:
`Dolog B:`

Step 2 Enter the Bsdol function
`RAMZU-SEAB`.

Step 3 Following the requests on the screen you define a new segment area beginning at address 0 of one segment:

Example:
After first use of PLC with AKF35 V6.x the allocation for RAMZU-SEAB is as following:
`15:18432, 4095`.
With following entry you redefine this area at the begin of segment 15:
`15:0, 4095`.

Note This change is **not possible during active** communication.

Step 4 Change the memory allocation defined during first use of PLC with the Bsdol function ASB (Change and Diplay Memory Area)

Example: (Entries are underlined)

```
DOLOG B:   ASB <RETURN>
NUMBER OF STORAGE AREA    :30 <RETURN>
STORAGE AREA 30(RAM) 15: 18433 TO 22528
SEGMENT: 15 <RETURN>
FRM: 1 <RETURN>
TO: 4096 <RETURN>
TYPE=RAM
STORAGE AREA 31      : E <RETURN>
```

React. Now you can use the free area beyond the end of the RAMZU-SEAB area for any other purpose.

III Modicon A500 with BSW V6.x and AKF35 V6.x Modification of Equipment List when using Intelligent Function Modules

Warning You only may modify the equipment list offline with AKF35.
An online modification with Bsdol function BES ”Enter and Change Equipment List” is not allowed, the PLC can crash in this case.

IV Reservation of Areas for Parameter Fields in the Cross-Reference List (AKF35)

Note If the length of a parameter field is not known exactly during configuration, the system reserves an area of max. length and will display this in the cross-reference list.

For fields whose length cannot be defined exactly or for which the field length is >255, only the start and possibly the end of the field are entered in the cross-reference list as single signal(s). Since the cross-reference list is incomplete in this case, a respective message is given.

Example: SFB No. 259 (AWE13)

The parameter WA is the first word of a field, in which the input of the measured values is stored. The number of channels is only computed at PLC runtime out of the parameter KA. The maximum value of KA is 255, that means a reservation of 255 maker words starting with address WA is done by the system.

D Belongs to software kit E-No. 424-271510

V Modicon A350/A500 with ALU 150 and AKF35, V6.x Program downloads into the PLC via quick PADT (P820C, P840C, IBM compatible PC with processor $\geq$ 486)

(05/94)

Caution When using a quick PADT and the menu item ”Load → Program to PC*” to download your AKF35 program into a PLC that is equipped with an ALU 150 central unit, the program will abort the transfer. The download operation remains locked in the function ”Initialization Equipment list front connection”, and the initial values are also not transferred.

Remedy

There are three alternatives to choose from:

- **Terminating the download operation after cancel**

  Proceed as follows to do this:

Step 1	Select the menu item ”Online → Terminal Mode”.
Reaction	The ALU 150 will be in the online function ”BES” (enter and modify equipment list).
Step 2	End the ”BES” function with the entry `E <Return>`.
Reaction	The screen will be show the prompt `Dolog B:`
Step 3	Exit the Terminal Mode.
Step 4	Transfer the initial values to the PLC through the menu item ”Load → Initial Value to PC*”.

❐ **Reducing the processor speed of PADT to perform the download operation as usual**
Proceed as follows to do this:

Step 1	Invoke the BIOS setup program of your PADT (refer to the PADT Operating Instructions / System Manual).
Step 2	Set processor speed to ”Low”.
Step 3	Set the systems cache to ”OFF” which will disable it.
Step 4	End the BIOS setup program by saving the modified settings (refer to the PADT User’s Guide/Manual).
Reaction	The PADT is now low enough to interact with the ALU 150 as usual.

☞ **Note** Return your PADT to its initial state when you have finished working with the AKF35/ALU150.

❐ **Utilizing a PADT with a slower processor (< 386)** for service with AKF35, Version 6.x, and Modicon A350/A500 with ALU150.

VI Counter and Timer in the same network: Malfunction in AKF35 Version 6.41

(12/97)

Problem

If you place the output of a timer onto the reset input of a counter in the same network, the counter will take the setpoint value of the timer as setpoint value rather than its own programmed setpoint value. For instance, if you entered a setpoint value of 10 for the timer, the counter will set its output after 10 pulses, regardless of the setpoint value indicated in its own makrer word.

Remedy

This malfunction will be prevented if you program timer and counter in different networks.

H Belongs to software kit E-No. 424-271510